Csdn.net 技术改变世界 创新驱动中国

Programmer 程序员

2009年 11月刊

邮发代号:2-665

■ 封面报道

WINDOWS 再现辉煌

结束了8年时间的等待，Windows 7终于发布了。微软如何通过这个产品再现辉煌？来自微软、英特尔、英巴卡迪诺、雅虎中国、支付宝以及金山的专家从不同角度为您解读Windows 7。



■ 高端视点



再谈如何成为技术领袖？

■ 精彩推荐

一个老拥趸眼中的极限编程之父

——记2009年Kent Beck中国行

又当爹又当妈的产品经理

俄罗斯软件业面面观

用三叉搜索树实现高效率的“自动完成”

价值驱动的组织转型

内附CTO专刊

【领导者】27年看IBM研发

【CTO说】技术管理之痛

【行业】小平台成就大市场

ISSN 1672-3252

9 771672 325098 11

程序员 Programmer 2009年 11月刊 技术改变世界 创新驱动中国

特别感谢

（排名不分先后，以姓氏首字母为序）

Contributors



Adam Anger

感谢微软公司大中华区Windows商用产品部总经理Adam Anger接受本刊采访，详见本期封面报道。



Kent Back

感谢极限编程之父Kent Back接受采访，并与我们分享响应式设计的观点，详见本期“报道”栏目。



戴晔

感谢风云在线总裁戴晔先生接受本刊采访，并介绍微软CRM模板设计大赛的情况，详见本期“报道”栏目。



邓毅

感谢网易有道技术总监邓毅分享有关架构方面的观点，详见本期“架构师接龙”栏目。



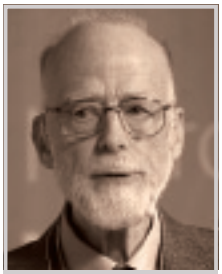
何伟平

感谢雅虎中国首席工程师何伟平先生为本刊撰稿，详见本期封面报道。



黄冬

感谢彼岸网联（北京）技术有限公司创始人黄冬为本期“架构师接龙”栏目提出问题。



Tony Hoare

感谢快速排序算法创始人、可验证软件理论泰斗、图灵奖获得者、微软剑桥研究院首席研究员托尼·霍尔（Tony Hoare）爵士接受本刊采访，详见本期CTO专刊。



琚伟

感谢英巴卡迪诺大中华区总经理琚伟接受本刊采访，详见本期封面报道。



寇卫东

感谢IBM软件集团大中华区总工程师寇卫东与本刊分享成为技术领袖所应具备的素养，详见本期“高端视点”栏目。



梁兆柱

感谢英特尔公司亚太研发有限公司总经理梁兆柱博士接受本刊采访，详见本期封面报道。



林知坚

感谢IBM杰出工程师林知坚博士接受本刊采访，详见本期CTO专刊。



John Lilly

感谢Mozilla全球CEO约翰·里利（John Lilly）接受本刊采访，详见本期“高端视点”栏目。



Mike Nash

感谢微软公司Windows产品部副总裁Mike Nash接受本刊采访，详见本期封面报道。



Andrei Pronin

感谢Auriga公司总经理Andrei Pronin为我们分享俄罗斯软件业的发展历程及现状，详见本期“报道”栏目。



Ramesh Rajagopal

感谢微软中国研发集团服务器与开发工具事业部部门经理Ramesh Rajagopal为本刊分享有关敏捷团队和软件开发的观点。



Eric Sink

感谢SourceGear公司创建人Eric Sink为本刊分享有关产品经理方面的观点，详见本期“实践”栏目。



王晓明

感谢阿拉丁敏捷科技（北京）有限责任公司首席合伙人、组织转型专家王晓明为本刊分享有关组织转型方面的看法，详见本期“实践”栏目。



韦建儒

感谢用友软件公司小型企业管理软件事业部技术总监韦建儒接受本刊采访，详见本期CTO专刊。



Robert Wahbe

感谢微软公司服务器与开发工具副总裁Robert Wahbe接受本刊采访，详见本期封面报道。



徐嘉

感谢Symbio（信必优）公司CEO徐嘉先生接受本刊采访，与我们分享关于外包企业发展的观点，详见本期“高端视点”栏目。



徐曦

感谢微软CRM解决方案经理、高级CRM顾问师徐曦先生接受本刊采访，并分享关于CRM的观点，详见本期“报道”栏目。



杨钢

感谢金山软件WPS首席架构师杨钢为本刊撰稿，详见本期封面报道。



张侠

感谢SAP大中华区CTO张侠接受本刊采访，分享清晰企业的观点，详见本期“高端视点”栏目。



赵善中

感谢结构行为合一架构方法创始人、SBC架构之父赵善中先生为本刊撰文，详见“评论”栏目。

卷首语

Foreword

做好准备，迎接 3G

自 1973 年第一通蜂窝电话打出开始，十年后进入 1G 时代，再十年进入 2G 时代，21 世纪初进入 3G 时代。在 2G 平台上的竞争大局已定后，中国主政者和运营商们看向了 3G。2009 年，3G 开始在中国商用。

从 1G、2G 到 3G，移动设备从单一通讯功能逐渐发展为一种强大的计算平台。MTK（硬件）与 Android（软件）把客户端的计算设备成本压到极为低廉的水平。当设备不再是暴利来源，通路与应用就变得前所未有地重要起来。Apple 开创了 AppStore 模式，Apple AppStore 上有 10 万开发者注册，将近 7 万应用带来 15 亿次下载。一时间，几乎所有厂商和运营商都开始上马应用程序商店项目。有些第三方公司也在做这方面的尝试。

3G 给移动计算平台带来的变化在于带宽的低成本增加和由此导致的永远在线可能性。因应全业务和智能融合趋势，这种变化将强烈冲击原有的移动业务模式。包括运营商、SP、CP 在内的传统受益者，不得不重新审视业务、跳出通讯想问题。而软件企业也将勇敢跨越电信与 IT 之间的壕沟，在面向个人、企业和政府的应用领域，大量引入移动计算特性。

既有利益分配方式将在相当一段时间内阻碍面向个人的应用全面爆发。幻想运营商全面开放通路和提供公平的移动应用行销竞争渠道，是不切实际的。企业信息化可能会是 3G 移动计算的第一战场，而且也会是影响极为深远的应用领域。

无论如何，移动计算进入 3G 时代，对于开发者而言是利好消息。计算平台的每一次革命，都会催生新的 IT 霸主。大型主机时代有 IBM，PC 时代有微软，互联网时代有 Google，而移动计算 3G 时代给开发者带来的机会则要比 PC 和互联网都更令人激动。数以亿计的无线设备时刻在线，意味着存在庞大的长尾市场。在这个市场上，有无数的小蓝海等待开发者去征服。可以预见，未来数年乃至十年内，新一代霸主将独占鳌头。

PC 和互联网的共同特征是计算平台和开发技术趋同。移动设备目前仍是五花八门，即便是同一品牌、同一操作系统的手机，互相之间也常常有软件兼容性问题。当年的 Symbian 联盟、现在的 Android 和中移动 OPhone 系统，都试图给出解决方案。未来谁成为标准，还很难说。平台的统一有利于应用的产生和推广，却是毋庸置疑的。

于 2009 年 10 月 22 日开幕的软件开发 2.0 大会上，移动开发是最重要的部分之一。抓住移动开发，就抓住了未来。对于软件企业，抓住开发者，就抓住了移动开发的未来。让我们准备好，迎接 3G 时代的到来吧。



Programmer 程序员 CONTENTS
技术改变世界 创新驱动中国 ● http://www.programmer.com.cn



42 Windows 7: 再现辉煌
44 第七代Windows
——再现微软时刻
46 Windows 7中的架构视点
50 客户端与服务器的对话
——Windows 7是新产品吗？
51 嵌入式，与Windows 7保持同步
52 谁为Windows 7买单?
54 Windows 7研发背后不为人知的故事
56 Moblin不与Windows 7竞争
58 Windows 7不是微软的革命
——英巴卡迪诺大中华区总经理瑞伟访谈
60 Windows 7 的推出与操作系统的变迁
62 Vista栽树，Windows 7乘凉
64 在Windows 7中寻找渐渐远去的WinFS

固定专栏 Columns

10 名人堂 Hall of Fame
20 程序天下事 Technical News
27 业界新闻 News
41 漫画与幽默 Humor

高端视点 Leaders' Viewpoint

12 再谈如何成为技术领袖
——技术不是充分条件
14 外包企业“合纵连横”之道
15 清晰企业
16 开源，也可以做出好产品





报道 Report

28 俄罗斯软件业面面观
32 南国热风，广东特色
——走进广东软件科学园
36 一个老拥趸眼中的极限编程之父
——记2009年Kent Beck中国行
38 风云再起之CRM
——记微软CRM模板设计大赛
40 掀起OPhone平台开发的红盖头

管理 Management

66 价值驱动的组织转型
由金融风暴引发的经济衰退正在慢慢消逝，各种各样的公司和企业也在逐渐恢复元气。然而幸存下来的组织们不能不考虑这个问题：我们应该如何改变自己，才能增强自身的生存能力，并具有持续可发展能力？我们应该如何改变自己，才能增大自己对客户的价值？我们应该如何改变自己，才能在当前环境下生存？
69 又当爹又当妈的产品经理
一个成功的产品就像一个人的成长过程一样，会经历孕育、婴儿、少年、青年、成年、老年等几个不同的阶段，而产品经理就像无数操心的父母一样，在不同阶段要有不同的心态和处理方式，以帮助产品茁壮成长。
73 BlackBerry开发经历：（下）
——PodTrapper项目的经验教训
本文上篇刊登于《程序员》10月刊。文章介绍了作者在开发第一个BlackBerry移动应用——PodTrapper过程中学到的经验教训，内容涵盖开发、定价、零售和市场营销等诸多方面。

架构 Architecture

76 架构师接龙：黄冬VS.邓毅

测试 Testing

78 自动化测试结构论
本文基于商业自动化测试工具的研发，分析大量的商业自动化测试工具的实现模式，以提高大家对自动化测试工具本身的认识与理解。

VSTS团队兵法专栏 VSTS Team Strategy

82 没有银弹 量体裁衣
——Visual Studio Team Architect团队的敏捷软件开发之一

Programmer 程序员 CONTENTS
技术改变世界 创新驱动中国 ● http://www.programmer.com.cn

P32

报道 Report

南国热风,广东特色
——走进广东软件科学园

从位于广州市中心的广东省政府出发，向东偏北方向行进23公里，进入幽静的广州科学城彩频路东段。在这条不太宽的路上，最引人注目的建筑群，就是广东软件科学园了。



用户体验专栏 User Experience

84 实现用户的三个情感诉求
——QQ2009用户体验设计谈

一分钟先生 Mr. One Minute

86 如何应对客户方人员变更?

互联网 Internet

88 用三叉搜索树实现高效率的“自动完成”
自动完成能提高网站的用户体验，但要高效率地实现这功能，却是一种挑战。

企业开发 Enterprise

90 Transact-SQL元编程
——用T-SQL生成T-SQL
在执行过程中动态生成简洁的SQL脚本，代替设计时硬编码方式的脚本编写，使相同的脚本可以不经修改应用到完全不同的场合。

94 在Spring下分布式事件通知服务框架的实现
在分布式系统中，分布在不同机器上的服务之间如何协同工作是非常关键的，事件通知则几乎是解决这个关键问题的不二选择。

系统开发 System

97 .NET安全系列二：权限的运用
权限是CAS中的核心概念。如何正确的运用权限，是影响.NET应用安全的关键。

移动专栏 Mobile

101 OPhone联网应用开发中的线程管理与界面更新
使用AsyncTask处理联网，播放大尺寸媒体文件等较为耗时的工作，不但执行效率高，也可以节省代码。

调试之剑 Debugging Sword

104 在调试器中品味计算机系统的睡眠过程（上）

图书 Books

108 半路出家，从化验员到程序员
111 代码猴子与童子军军规
112 新书上架

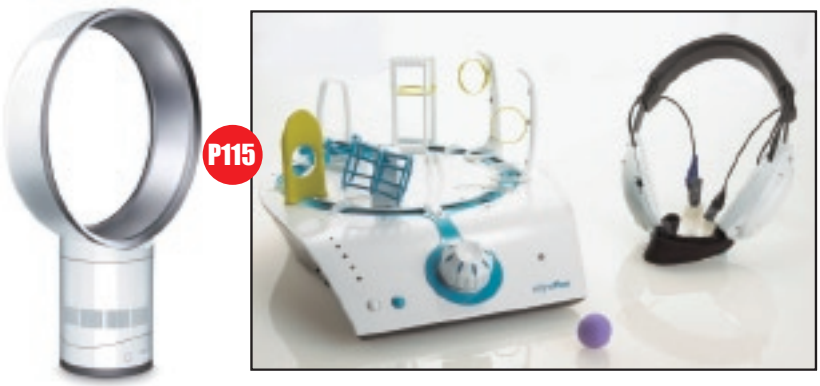
产品推荐 Products Recommendation

114 Geek产品
116 新产品 & 新工具

评论 Comments

118 软件的性格
120 “SBC架构”解答了软件规格的迷思





Contents

http://www.programmer.com.cn

42 Windows 7: Rebuild the Glory

After 8 years waiting, Microsoft released the latest version of their client operating system - Windows 7. For the failure of the last version, Windows Vista, how could we realize Windows 7? Could Microsoft rebuild the glory of the software empire? Only time would prove if the Windows 7 will get the success on the market nowadays. In this feature, we've invited several expert to talk about Windows 7 and client operating system in the future.

69 Product Parenting

Just like a normal human being, a successful product goes through pregnancy, infancy, childhood, adolescence, adulthood and old-age. A responsible product manager will act like regardful parents, and he shall be prepared with right mindset and approaches at right stage.

66 Let Value Drive Organization Transition

After analyzing the recent global and local economic data about various aspects,many analysts and observers make the conclusion that the economic depression caused by financial storm begins to vanish. Feeling lucky deep inside, the survivors cannot help wondering: in what way we should change ourselves to strength our core competences, increase our values to our customers, and manage to survive again in the next storm in the future?

88 Efficient auto-complete with a ternary search tree

Auto-complete definitely makes for a nice user experience, but it can be a challenge to implement efficiently. In many cases, an efficient implementation requires the use of interesting algorithms and data structures.

90 Transact-SQL Meta-programming

SQL Server 2005/2008 provides meta-data in catalog view, allowing us to dynamically generate and execute SQL scripts. Such technique can be used to solve some complicated problems in database management.

《程序员》杂志招聘启事

《程序员》杂志社以求贤若渴的态度欢迎优秀人才的加盟。

招聘职位：记者/技术编辑 各2名

岗位职责：

1 《程序员》杂志及相关内容出版物的选题策划；
2 负责栏目的组稿、采访、制作等编辑工作；
3 联络并拓展杂志的专家资源，不断优化内容结构；
4 掌握杂志出版流程，并跟踪完成。

要求：

大学本科学历，计算机相关专业；
熟悉常见编程语言和工具，对目前主流技术有一定的认识和理解，对主流及前沿的软件技术发展具有高度的敏感性及热情；
良好的中、英文写作和翻译功底；
具备较强的沟通及表达能力；
工作认真敬业，较好的团队合作能力。

特别优秀者，可不受以上具体条件所限。

联系信箱：hr@csdn.net (请在主题栏注明“程序员杂志”)，欢迎联系！

欢迎加入我们，与中国软件产业同成长！
与《程序员》共筑辉煌！

Programmer 程序员

2009年11月刊 总第205期

主管：中国社会科学院
主办：中国社会科学院文献信息中心
出版：《程序员》杂志社
网址：http://www.programmer.com.cn
国际刊号：ISSN 1672-3252
国内刊号：CN11-5038/G2
邮发代号：2-665
广告经营许可证号：京东工商广字0188号

总编：黄长著 Editor-in-chief：Huang Changzhu
社长/常务副总编：张悦校 President：Zhang Yuexiao
副社长：蒋 涛 Vice President：Jiang Tao
编委会：黄长著 张悦校 陈洋彬 蒋涛 曾登高 韩磊
Editorial Member：Huang Changzhu Zhang Yuexiao Chen Yangbin Jiang Tao Zeng Denggao Han Lei

执行主编：孟迎霞 Executive Editor-in-chief：Meng Yingxia
编辑部主任：孟迎霞（兼） Director：Meng Yingxia
编辑部副主任：欧阳璟 Deputy Director：Ouyang Jing
责任编辑：郑柯 李雨来 郭晓刚 丁莹
Editors：Zheng Ke Li Yulai Guo Xiaogang Ding Ying
特邀编辑：方梁 高昂 常政 赵健平 吕娜 卢鹍翔
Contributing Editors：Fang Liang Gao Aang Chang Zheng Zhao Jianping Lv Na Lu Dongxiang
美术设计：纪明超 Art Designer：Ji Mingchao
美术编辑：吴志民 Art Editor：Wu Zhimin
Tel：010-64351458
Email：editor@csdn.net

发行部 Distribution Dept.010-64351431
Email：sales@csdn.net

广告总代理：北京创新乐知广告有限公司
Sole Advertising Agency：Bejing CSDN Co.,Ltd
Tel：010-64376055
Email：ad@csdn.net
Marketing Dept：010-51661202（ext 149）
Email：market@csdn.net

读者服务部
Readers service Dept.
网上订购：http://book.csdn.net/programmer/
读者信箱：reader@csdn.net
地址：北京市朝阳区酒仙桥路14号兆维工业园B区3楼2门1层
Address: B3-2-1F, Zhaowei Industry Park, No. 14 Jiuxianqiao Road, Chaoyang Dist, Beijing
邮政编码：100015
电话：010-64351436
传真：010-64348545

法律顾问：北京中润律师事务所 王杰
Law Consultant：Beijng Hengsheng Lawyer Firm
印刷：北京盛通印刷股份有限公司
Print: Beijing Shengtong Printing Co., Ltd.
出版日期：每月1日
Publication Date：the first day per month
零售价：RMB 15.00元 新台币 390元 HK $ 35.00（港、澳）
US $ 9.00（海外）
Retail Price：RMB 15, NT$390, HK $ 35.00,US $ 9.00

本刊文章版权所有 未经许可不得转载
发现装订错误或缺页，请将杂志寄回本刊读者服务部，即可得到调换。

开拓梦想之地的不老顽童

——微软共同创办人保罗•艾伦



■ 文 / 吕娜

世界首富比尔•盖茨的名字家喻户晓，另一个掩藏在这耀眼光环之下的名字却容易被忽略，那就是微软的共同创办者保罗•艾伦(Paul Allen)。有人说，没有保罗•艾伦，微软也许不会出现，是他将盖茨引入软件这个行业。

保罗•艾伦1953年出生于西雅图，父亲在华盛顿大学图书馆当过20多年的图书管理员，为艾伦接触到更多计算机相关的图书创造了便利条件。艾伦14岁时已经极为痴迷计算机，整天泡在学校机房里玩弄设备并学习编程。1968年，他在湖滨中学遇到另一个深度计算机迷——年仅12岁的比尔•盖茨。两人很快成了好朋友，整天在一起大谈计算机编程，相互比试。

1971年，艾伦考入华盛顿州立大学，但在两年后毅然退学。理由很简单，他希望实现自己软件创业的梦想。艾伦全力鼓动盖茨一起创业，盖茨回忆说：“他老是说，再不干就迟了，我们将遗憾终身。”但当时盖茨并不想因为荒废哈佛大学的学业而令家人担忧。

1974年12月，艾伦看到新出的《流行电子》上刊登了世界第一台微机Altair8800，他拿着杂志兴奋地跑去找盖茨，盖茨终于被说服，在随后的两个月里，两人没日没夜地为Altair8800开发Basic计算机语言，很快大受欢迎。“有时决定你一生命运的就在于结交了什么样的朋友，”盖茨在回忆中这样描述，“当时如果不是艾伦描绘的蓝图打动了我，也许我还会待在大学里。”

1975年，他们共同创立了“微软帝国”，两位创始人配合十分默契，艾伦专注于新技术和新理念，盖茨则以商业为主。艾伦富于远见，他想的都是10年以后的事情。早在10多年前，艾伦就说过：“信息高速公路将是下一波潮流，你不能经常赶上两波这样的潮流。”

1982年艾伦因病离开微软，恢复健康后并没有回微软继续工作，而是开始了投资生涯。20世纪90年代初期，他开始思考宽频的未来及其形式：到底是通过有线电视、卫星还是Internet。于是在信息高速公路、多媒体、有线电视以及数字电器、音乐、航天科技等市场频频投入巨资，被媒体捧为高科技最活跃的投资者之一。他本人投资掌控的公司和组织甚至超过150家，每年都有30笔以上的投资，平均每笔500万美元左右，几乎涉及各个领域。

和盖茨一起创建微软，无疑是艾伦一生中做得最成功的一项风险投资。但艾伦后来的投资往往不按理出牌。人们评价，他的投资像抛一把骰子般随意。更有甚者，《福布斯》尖锐地指出，“艾伦最大和唯一的成功就是与盖茨共同创立了微软公司”，他在微软之后的商业活动多半没有取得可观的回报。但艾伦并不认同这种评价，他很享受目前的生活。与盖茨的雄才大略相比，艾伦的确算不上是一个出色的管理者或经营者。但正如艾伦集团的高级经理比尔•萨沃里所说，这里确实有一种“艾伦效应”，即一旦艾伦投资某个领域，这个领域就会变得很有价值。也有人这样猜测：这些投资看似随机，却各自拥有其根基，逐渐构架成在线世界的艾伦帝国。

在艾伦的身上有着多重标签和属性：商界巨子、富于幻想的开拓者、一掷千金的豪客。他对天文科学和“外星人”的痴迷程度令人叹为观止。除了是“宇宙飞船一号”的发展资金支持者，2004年他旗下的SpaceShipOne小型人造火箭，还创造了全世界首个私人飞行装置攀上离地112公里高空的纪录。艾伦还投资建立了一座“外星人博物馆”，捐资2500万美元用于寻找外星生命。艾伦还是一个老顽童，他作为超级篮球迷，拥有西雅图海鹰美式足球队以及波特兰开拓者篮球队。作为超级音乐迷，他自组过两支摇滚乐队“肉店男孩”和“成年男人”。2000年，后者推出的首张同名专辑中，艾伦担任吉他手，据说很是专业。

“你还拥有梦想吗？”据说这是艾伦招聘私人助理时面谈的一个问题。艾伦五花八门的投资，更多源于他的各种梦想。艾伦和他的人生，就像他旗下一家电影公司的名字——梦工厂。就像他写的一首《时间炸弹》歌词里唱到的：“我所做的一切也许都是错误的，但我仍然会斗志昂扬。”艾伦仍然一如既往地做着自己喜欢的事，不断创造一个又一个传奇。

再谈如何成为技术领袖
——技术不是充分条件

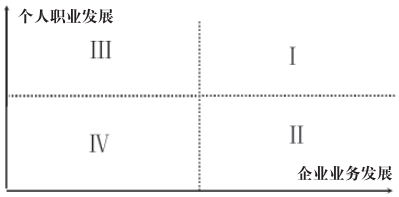
要想成为一名优秀的技术领袖，需要具备哪些方面的素质？我在 IBM辅导很多年轻同事，如何才能帮助他们在职业生涯中成长为技术领军人物？是否只要拥有了过硬的技术本领，就能成为技术领袖？许多年轻的朋友技术拔尖，然而却觉得没有得到领导的赏识，这又是什么原因呢？我常常都在思考这些问题。去年这个时候，我同读者朋友们分享了一些心得，我想结合自己多年的高校教育和 IT从业经验，再同大家谈谈这一话题。

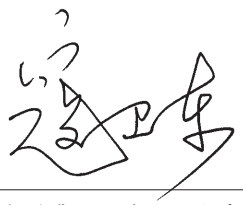
1. 我很清楚地告诉大家，要成为技术领袖，技术是必要条件，但不是充分条件。如果一个人技术基础不够扎实，他几乎等同外行，领导其他技术人员不说没有可能，但至少会相当困难。另一方面我们在国内外都能看到不少拔尖的技术人员，有的是一流的科学家，有的是一流的技术专才，可一旦把他们放在领导岗位上，却都以失败告终。他们虽然技术出色，但是没有当领袖的能力，工作中四处碰壁，缺乏凝聚力，不得人心。



2. 要成为技术领袖就要清楚公司的业务，知道技术应当如何配合业务需求。作为技术领袖，只有做到这点，才能使公司业务有大的发展，才会得到高层领导的认同。如果你只懂技术而不懂公司业务，或者你在大学工作而不懂领导教学科研，是没有办法胜任技术领袖的。我们钻研技术的朋友往往忽略了这一点，对某项技术非常专注，但是两耳不闻窗外事，没有把注意力放到公司的发展需求上来。

为了更好地协调个人职业发展与企业业务发展，需要处理好工作的优先级，分清主次。在日常工作当中，我常常为同事画下面这幅图，帮助他们进行分析。





IBM 软件集团两岸三地大中华区
总工程师 寇卫东

如果有一项工作摆在面前，我们如何定义它的重要性，应该花多少力气去做？我们可以通过象限分析找出答案。第**I**象限表示此项工作对企业业务发展和个人职业发展都很重要，这是最完美的结合，优先级应该放在最高级。我们在企业中应该多做此类工作。第**II**象限表示此项工作对企业业务非常重要，但是对个人目前的职业发展看似并不特别重要。我们要以企业的利益为重，所以优先级应该放在第二位。第**III**象限表示此项工作对个人职业发展很重要，但是对企业业务发展不那么重要，我们应该把它放在第三位。第**IV**象限表示此项工作对企业业务发展和个人职业发展都不重要，优先级应该是最低的。对这类工作，自己最好不做或者少做；就是做，也应该尽可能少花时间去做；或是请其他同事去做，因为对你的个人职业发展不重要的工作不等于对其他同事的个人职业发展也不重要，换言之，此项工作可能对其他同事的个人职业发展很重要呢。

许多企业都在创造一种文化，希望能将企业业务发展和员工的个人职业发展紧密结合起来，尽管有时候很难兼顾。作为一名领袖人物，他一定更多地在关注大局，而不是仅仅看到自己。关注大局的人会处处从公司的业务、公司的需求出发。在实际工作中我们常常可以看到，一个真正把大局放在第一位的人，即便今天不是领袖，明天也一定非他莫属。现在的领导不欣赏，将来必会为其他领导所赏识。

3. 一个能把握大局的人需要具备很多能力，其中非常重要的一项就

是我下面要谈到的第三点——决策力。作为技术领袖，要有坚强的领导能力，敢于面对各种困难的挑战，敢于决断。有些人天生就有准确的决策能力，而更多人需要靠后天来培养。

在我的孩子还小的时候，我就给他灌输这种思想，培养锻炼他。如果他要问我，我就让他自己做决定，不论决定是对是错、是好是坏，在我看来都是正确的。因为最大的错误是该做决定时你举棋不定。再举一个例子，美国前总统布什刚刚上任的时候，很多人批评他没有总统的风范，但是经过几年的洗礼他的确拥有了总统的气质。原因又是什么呢？因为那个职位训练了他，很多情况他必须要做出决策。

对于程序员来说，虽然无法像领导那样做重大的决定，但仍然可以在自己的范围内做出决策。例如使用什么样的技术手段，采用何种算法实现，选择哪种数据结构，如何撰写文档，同谁一起合作，这些并不一定非要等老板或者构架师来判断。大家应该有意识地锻炼自己，慢慢成长，直到将来能够领导大的项目。每个人不可能一开始就能对一个完整的系统做出决策，可是如果不从当程序员的时候就开始锻炼自己，将来也做不了。

面对决策失败可能带来的后果，我们应该学会坦然面对，学会承担。如果不去做决策，不愿承担责任，就永远无法取得进步，无法获得做技术领袖的能力。

4.要想成为技术领袖必须有远见，有智慧。有远见有智慧的人才能有承担。我有一个在加拿大的同事，开始的时候只是一般的技术人员，很短的时间内就变成经理，再后来没有经过多长时间又被提升为主管。原因很简单：他有远见，并且有足够的智慧知道应该如何去做。像他这样的员工虽然刚开始的职位很低，但是公司领导觉得这样的人才对公司将来的发展大有帮助，另外，如果这样的员工离职，不但是公司的损失，还会助长竞争对手的实力。所以公司一旦有职位就会提升这样的员工。我们作为技术人员也是一样的道理，面对一套大的系统方案，只有穿透障碍看到未来，才有机会获得提升，成为技术领袖。

5.成为技术领袖需要自信、自觉和自律。自信，这和领导能力有关，没自信怎么来领导别人呢？自觉包含两个方面的内容：自我感知和自我解嘲。自我感知是要对自己有一个正确的评价，“人贵有自知之明”——我们老祖宗都这样讲。有些人不知道自己的情况，要么估计过低，要么估计过高，对自己没有一个公正的评价。很多事情要靠下意识的感觉，你可能并不知道具体发生了什么事情，但是凭感觉就能判断出对错。自嘲是能给自己台阶下，这很重要，往往很多人都做不到，一不小心就陷入骑虎难下的境地。另外对自己有一个正确的评价，才不会弄得自己下不来台。

自律就是要约束自己，保持良好的品行。自律不是做给别人看，也不是做给领导看，而是为自己而做。每个人心中都应该有把标尺，衡量什么事情该做，什么事情不该做。自律很关键，年轻人尤其应该加以重视。有好的人品才会得到大家信任，如果不自律，没有良好的品行，有谁会对他心悦诚服，又怎样来做大家的榜样呢？

6.勇于承担，敢于付出。也就是说一个人的工作既不是为了金钱，也不是为了名誉，单单只是为了实现自己人生的价值，获得成功。自己驱动自己，不畏艰难险阻，不达目的誓不罢休。不愿意承担，不愿意付出，处处计较得失的人是没有办法成为领袖的。

这里所说的成功，并非单纯的职位上的成功，还包括技术上的成功，做人的成功等。其中做人的成功最重要，往往却被最多的人所忽视。职位只不过是领导给的一个头衔，技术领袖所具有的影响力并不是单单靠领导所给的头衔就能获得的。不管大公司小公司，很重要的一点就是先做人。我认为衡量一个人是否成功，首先看做人，然后看技术，最后才是职位、名誉、金钱。而且往往人和技术做好了，金钱名誉就会随之而来。正如诺贝尔奖得主们无一人不是勤勤恳恳做事，假使一开始就想着名誉、奖金，或许他们永远都无法取得举世瞩目的成就。

成功是一个很宽泛的概念，如何衡量成功是因人而异的。一般来说，成功是一个正态分布的频谱，不是一个单脉冲。有些人拥有很高的职位、很丰厚的收入，但却没有家庭、没有生活，因为他们把所有心思都放在了工作上，最终获得了事业上的成功，一个脉冲式的成功。相对他们的度量标准来说是成功了，然而对另一些人来说，他们并不成功、或者不值得效仿。因为事业成功只是一方面，除了工作，我们还有生活，还有家庭的维系、子女的教育、八小时后的个人爱好。一个和谐的成功是一个频谱式的成功：也许没有哪一方面是特别突出的，但是有几个方面是相当好的，其他方面也还可以，这样综合来看是最好的。

7.能领会别人的感觉和想法，会与他人相处。一个拥有领袖潜质的人，往往能够很快感知周围的氛围，别人的感受，他能够通过简单的语言、位置调整，使大家轻松舒服地投入工作。

8.能把大家团结起来，人气旺。很多时候，某个人并不是领导，但是大家都愿意听他的话，成了无冕之王，这样的人即便现在不是，将来也必然会成为领袖，这只是时间的问题。

读者朋友们大多都是技术人员，但是大家不要被技术所束缚。我年轻的时候也走过弯路，希望这些心得体会能对年轻的朋友们有所帮助。成为技术领袖并非遥不可及，只需要比别人多付出一点坚定和执著，多注重先做人后做事。

外包企业“合纵连横”之道

回顾外包产业刚刚踏入国内市场时，中国以其低成本，人力资源丰富和极具潜力的市场等特点，吸引了众多外包企业的加盟。而如今，在经过爆炸式增长后，整个外包业进入了一个瓶颈阶段。大部分外包企业就如同战国时期的各诸侯国，想要在日趋紧张的形势下不“坐以待毙”，就要寻求新的发展战略。

中小型外包企业——寻求并购伙伴

国内一些中小型外包企业由于规模小造成的种种问题，直接导致了企业发展上的停滞不前。因此，要想解决这个问题并真正发展壮大，企业整合是必不可少的。在整合过程中，当一个企业的资产、技术、市场缺乏竞争力时，寻求并购伙伴不失为一则上策。这虽然会产生一部分的权利让渡，但是从长远看来，并购可以让企业从规模上得到扩充，而外包企业规模大不仅意味着整体的运营成本可以降低，还代表着企业营运大项目能力的提高。



曾经作为外包进入并购时代标志的“东软与华信并购案”这一事件，虽然最后没有成功，但我们可以推断并购达成后可以带来的效应：从根本上可以改变中国外包企业的竞争格局；在技术上的融合可以赢得更广阔的外包市场，得到更多国际客户的青睐；而在追赶印度的脚步上更无疑像是安装了一台加速器。

大型外包企业——寻求合并盟友

在金融危机的冲击下，对于许多外包企业来讲，当市场发生变化时，因没有办法控制自身市场和没有核心技术产生的弊端造成了企业发展停滞不前甚至萎靡不振的局面。因此，对于这些外包企业来讲，合并产生的协同效应会给这些企业带来双赢。

在进行外包企业合并的时候，并不仅仅再是为了两个或多个企业规模上的单纯扩大，与并购有所不同的是：要从外包专业上做到根本的整合。就Symbio（信必优）合并案来看，这不仅仅是人力上的优化，它实现了跨技术、跨地区及跨行业的扩展，并且带来整合的技术和解决方案，对于需要资源来满足开发和技术整合需求的大型企业来说，合并实现了各方共赢。因此他们期待为全球企业客户，提供全面和创新的解决方案。如此一来，外包企业的竞争力会在合并中得到前所未有的提升。



信必优首席执行官　徐嘉

所谓外包企业间的合并，就是要将各外包企业的战略重组，在技术上做到优势互补，从而提高竞争力。若想长久的深入国际外包市场，在合并伙伴的选择上也很重要：合并伙伴必须是对于欧美市场、客户的需求非常了解，掌握国内稀缺的技术和专业的企业。由于国外客户需要大量国际化的专业和技术来提供完善的解决方案，中国目前在创新上做的又还不够成熟，而国外外包企业在这些专业和技术上都非常领先，在这种情况下，国外高端的技术加上国内开发低成本的成长能力就变成一种合作的趋势。

企业发展须和人才培养双管齐下

中国虽然每年都有着大量从高校毕业的专业技术人才，但与印度这样成熟的外包同行比起来，数量上或许远远超过了印度，但在素质上的差异却是显而易见的。整合之外，出于对客户需求的满足，与 IT培训机构和国内高校进行合作，通过提供良好的培训机会来提高高校 IT专业人才的思考能力和设计解决方案的能力，让他们了解国外客户的需求，从而为企业提供连贯的、可信赖的解决方案。Symbio（信必优）在本次合并的同时，就积极推动与国内高校发展双赢的合作伙伴关系，一方面满足了高校大学生对实践经验的需求，另一方面又解决了未来开发过程中会遭遇的人才紧缺的态势。

在遭遇金融危机之后，国内外包企业的发展虽然遇到了前所未有的瓶颈阶段。但我们不能忽略的是，这也正是一个走可持续发展的机会。而外包企业间的“合纵连横”恰好为企业竞争力的提升和转型提供了很好的策略。

清晰企业

世界经济局势动荡不安，变化之快超出所有人的想象，每个企业都面临更加严峻的挑战。当前，企业要应对彼此对抗的两重压力：一方面来自日益增长的商业复杂性，尤其是基于全球商业网络的运营；另一方面是日渐加强的企业约束力和责任，全世界的股东都希望企业发展具有可持续性。

面对这些压力，企业必须清晰地洞察、思考和行动。唯有具备可视性，能有效、果断采取行动的企业，才能重新调整业务策略，优化运营流程；才能让企业对各个股东负责，维护公司的品牌和声誉，使企业处于更有竞争力的地位并最终胜出。

因此，我们概括了企业管理中的一些关键内容，并结合金融危机后的现状，在全球范围内提出“清晰企业”这一概念。简而言之，当今的最佳运营企业必须是一家清晰企业。清晰，能够提升企业管理全球业务的能力，可以帮助企业达成短期目标，更有助于长期可持续发展。

一家清晰企业，应当做到以下三点：首先，透明且有能力负责。更高的企业透明度能让领导者准确评估资产流动和现金需求，并且有利于企业内部的权责分明。所有领导各司其职，才能把企业的各项措施落到实处。责任明确，就可以提升企业所有员工的责任感。其次，要做到精益而灵活。企业必须敏捷高效，不能臃肿，不能人浮于事。管理层必须要留意新出现的威胁或机遇，并针对新的条件快速调整战略，迅速地响应客户需求。最后，要以客户为中心，并且具有协作性，能与所有合作伙伴很好地协同。大家开诚布公，配合才能更加紧密，合作的层次才能更加深入。

这些做法都很有好处，但问题在于，究竟如何才能提升企业的清晰度和透明度？答案是，从现在开始主动出击，提升洞察力、效率和灵活性。

缺乏洞察力，领导者便不能有效决策，从而影响业务绩效。如果不能准确及时地获得相关信息，就无从评估风险和制定策略，无法确保业务的持续健康发展。想要更透彻地理解风险，改进决策流程，企业必须主动采取行动，力求提升对业务和整个商业网络的洞察力。清晰企业深具洞察力，可以悉知流动资产和现金需求，最大化储蓄机会，并在运营过程中始终保持健康良好的资产负债和经营状况。

低效的业务流程会让成本增加，产生不必要的浪费，最终降低企业的竞争能力，阻碍企业的可持续发展。管理者需要分析企业的产品线，确定最佳资源分配，以提高盈利能力，还需要分析核心业务流程，找到节省运营成本的机会，更好地利用现有投资。

在变化迅速的商业环境中，如果企业缺乏灵活性就将阻碍其为消费者创造价值，同时也会减弱企业的长期竞争力。若想更有效地响应环境变化和客户需求，企业需要找到方法读懂消费者的需求信号，并在业务流程和业务网络中融入新理念。运营层面更高的灵活性，还有助于企业更有效地改进流程，从而消除浪费，并建立可持续的价值链。



SAP 大中华区 CTO　张侠

许多企业都通过削减成本和采取一种“等等看”的姿态应对当前的经济环境。这看似是一种合理的选择，却忽略了开拓创新、谋求可持续性发展的机会。可持续性发展是企业全面管理经济、社会和环境风险，以及提高盈利的能力。往往在经济衰退时期，推出新的品牌和产品更有机会为客户提供更多的价值，为企业带来更多效益。因此，面对不利环境，企业更应当主动采取措施，预测风险，发现机遇，力争获得可持续性发展。

在经过长期快速发展之后，我国许多产业都面临着结构重新调整。很多产业正在朝着自主产品、自主技术、自主专利和绿色环保、节能减排的方向发展，因而比以往任何时候都更需要提升内部管理水平、整合外部资源、加快产品创新速度以提高竞争力。

此次金融危机赋予企业一个难得的转型、升级机遇，使其可以调整前进步伐，明晰前进方向。通过实施清晰企业战略，积极行动，必将成就下一轮跨越式发展。



开源，也可以做出好产品

在当前的IT产业格局之下，如果希望做出一款优秀的软件产品，可能就意味着大量的资本和人力投入。不仅如此，在软件发布之后，还要花费大量的成本对其进行维护和更新。

其实，这并不是唯一的途径。有一些软件，利用开源社区和开源组织，吸引了世界上成千上万的志愿者参与其研发和完善过程。以Mozilla社区的开源产品Firefox（火狐浏览器）为例，迄今为止已经发展了十多年的时间，我们的宗旨是致力于提升互联网的体验，但是在这十多年的时间里，不仅吸引了全球225名顶尖的技术人才，还吸引了数千名志愿者参与浏览器的研发和完善过程。



这些来自世界各地的技术高手们并非为了金钱回报，那么究竟是什么样的力量吸引了他们呢？我想，这些人的心目中都有一个共同的信念，就是怎样才能把互联网世界和我们现实的世界变得更好。对于他们之中的大部分人来说，在能够基本保证自身的生活质量之余，能够有钱养活自己就已经满足了，他们希望将其余的力量都发挥出来用于改变世界。

当然，这些都还是志愿者们主观的想法，作为开源组织的组织者和核心成员，我们更致力于为这些志愿者营造一个完全开放、自由、创新的环境，让他们能够在这个环境里充分发挥自己的才能。当我们凝聚了诸多技术高手智慧的优秀产品诞生之后，我们既不为牟利，也不将垄断作为我们的目标。现在网络世界变得更好了，我们要做的就是创新，突破传统的框架来创造出更好用的软件产品，给用户更多的选择，这才是正确的解决之道。

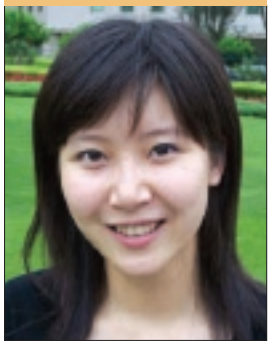
依旧以Firefox为例，我们的目标并不是让Firefox百分之百把浏览器这个市场吃掉，只是希望打破少数浏览器垄断的局面，能将网络变得更加开放。我们就好像是在市场上钻了一个洞，给其他的浏览器也提供了更多的机会来发展，致力于将互联网的质量提高，希望能够使更多的人们选择他们觉得合适的浏览器来满足网络需求。当然，我们现在有很多强有力的竞争对象，除IE浏览器以外，还有Google的产品以及其他具有知名度的浏览器。总的来说，这种现象反而体现了互联网对我们来说是越来越重要了，人们正在不断地从互联网上获取更多的东西。互联网在更多地介入我们生活的每一个领域、每一个细节里。

此外，值得注意的是，开源产品除了凝聚了大量创新的力量，在很大程度上会更加稳定和安全。因为，全球范围内的志愿者都会参与产品的测试和验证。他们可以帮助我们在特定的区域里面，给予一些使用的反馈，根据不同的使用和功能来提供不同反馈和建议，同时还会帮助我们记录一些不经常发生的崩溃现象，并且能够提交给我们对于一些特定环境下的反馈结果。这样的测试和维护流程，将比某家软件公司有限的测试人员测试出的结果更为可靠。

开源，也可以做出世界一流的好产品！



Mozilla 全球 CEO
约翰·里利（John Lilly）



陈阳 (Emily Chen)
GNOME.Asia
网站主编

关注GNOME亚洲峰会

一年一度的GNOME亚洲峰会按照惯例每年会选在亚洲的一个城市举办，第二届的GNOME亚洲峰会将于2009年11月20日至22日在越南的胡志明市召开。500名来自10个国家的GNOME领袖和开源贡献者们将共聚一堂。峰会将会为GNOME的用户、开发者、基金会成员、社区贡献者、学生、政府及企业提供一个平台，向大家介绍最新的GNOME技术以及GNOME在亚洲各地不同社区的状况。这也是首次在越南召开的国际性的开源会议。

这次峰会的主题是“Get Freedom with GNOME”，意为跟GNOME一起获取自由，跟随GNOME的引领者们一起去探索自由和开放源代码项目。这也揭示出GNOME亚洲峰会的首要目标：在亚洲传播GNOME的知识和自由开放源代码软件的理念，并在此基础上建立一个充满活力，繁荣的社区。 此次峰会得到了来自Sun，IBM, Google和越南政府的大力赞助和支持，各个开源社区也积极参与，如Debian，Ubuntu，Fedora，OpenSolaris以及LXDE社区的演讲者们将会在此次峰会上展示他们最新的技术和项目。本次会议讨论的热点将围绕着GNOME 3.0、国际化与本地化、移动平台、GNOME应用开发及桌面部署和教育软件等主题开展。演讲的形式也非常丰富，除了主题演讲，闪电演讲，还有动手实验和展示。如果你不能亲临现场，还可以通过实时峰会，Twitter，Facebook等网站随时关注峰会的最新情况。这样一个亚洲盛会，你绝对不可以错过。



陶震
Sun 中国技术社区
网站主编

使用 Policy Agent 保护应用程序

OpenSSO是一个由Sun发起的开源项目，专为 Web应用程序、单点登录、联合身份认证服务提供包括安全性在内的核心身份功能。该项目是在Sun Java System Access Manager的基础上开发的，构成了Sun OpenSSO Enterprise商业发行版的基础。与基于社区的OpenSSO子项目一样，它与SAML 2.0、XACML和WS-Federation等侧重于企业的标准共同为OpenID和Information Cards等协议提供支持。

OpenSSO服务器为验证、用户会话管理、策略管理、审计和日志记录等功能提供身份存储库和许多服务，它通常采用集中化的管理方式。结合Java EE Policy Agent，您可以通过OpenSSO的安全服务来保护Web应用程序，限制未经验证的访问。

通过在应用程序服务器实例上安装Policy Agent并对其进行配置，您可以强制执行验证、单点登录和授权。在安装过程中，为已部署应用程序提供保护的各个类将被添加到您的应用程序服务器实例中。在此之后，您还可以启用Web服务安全，为用户个性化应用程序，并映射到Java EE安全机制上。



霍泰稳
InfoQ中文站
主编

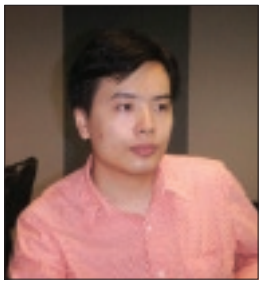
以退为进，IntelliJ开源

对于集成开发平台(IDE)领域来说，10月16日IntelliJ IDEA发布开源社区版是一个爆炸性的消息。

衡量一个产品是否成功，一般需要看三个条件：产品质量、用户群大小和是否赚钱。对于IntelliJ来说，产品质量没有问题，一直是业内公认的最佳IDE，如Eclipse等中的很多功能皆“抄袭”自它。至于是否赚钱，相信凭售卖授权协议养活几十个人也不是问题。关键在于其用户群的大小。这个闭源时代曾经的“大佬”，在Eclipse、NetBeans等开源IDE的冲击下，市场占有率逐渐萎缩，特别是低端市场。短期来看这不是问题，但长远来看，IntelliJ会逐渐进入无源之水的境地，到时再中途改变用户的使用习惯，说服他们从其他IDE转向自己就会变得十分艰难。

需要指出的是，这次JetBrains开源的并不是其全部产品，严格来说其开源的只是一个纯Java开发环境，如果你想做基于Ruby、Python等语言的开发，或者想和像Team Foundation Server、ClearCase这样的商业版本控制工具集成，你还是需要购买其收费的旗舰版。所以说，IntelliJ的开源，貌似迫于开源社区的压力，实则是一个很好的以退为进的产品策略。

微软技术



■ 主持人：Kaneboy
涂曙光，微软（中国）有限公司产品技术专家，博客堂成员。专注于.NET开发，从事Office System、SharePoint等产品相关的技术支持。

.NET开发工具的“少数派”：Delphi Prism与MonoDevelop

最近Embarcadero公司（Delphi的新东家）发布了Delphi Prism 2010。不要把它和Delphi 2010搞混，Delphi 2010仍旧是作为一款Windows本地原生应用程序开发工具而存在，而Delphi Prism 2010则是Embarcadero为.NET/ASP.NET开发人员所提供的一款.NET平台开发工具。对于熟悉Delphi语言语法、希望能在Visual Studio这个主流开发工具中使用Delphi语言和开发.NET平台应用程序的人来说，Delphi Prism 2010无疑是一个非常具有吸引力的选择。Delphi Prism 2010可以以一个大插件的形式，嵌入到Visual Studio Shell（一个未内置任何语言开发支持的Visual Studio“外壳”）、Visual Studio 2008，以及还未发布正式版的Visual Studio 2010中。

Delphi Prism 2010的另一个重大优势，就是它支持开发跨平台的应用程序。没错，虽然.NET平台本身只能运行在Windows操作系统之上，但不要忘了还有Mono呢。由于Mono，这个.NET的开源实现，能够支持Linux和Mac平台，而Delphi Prism 2010让自己产生的应用能够运行在Mono平台之上，于是乎，使用Delphi Prism 2010作为开发工具的开发人员，将发现创建的应用程序能够部署到Windows/Linux/Mac多个操作系统之中。Embarcadero公司依靠这一点，无疑将吸引到不少既希望能够基于.NET平台开发应用，又希望能使用一个功能强大的开发工具，同时还希望最后生成的程序能跨平台运行的开发人员和软件商。

说到Mono平台，实际上Mono项目一直也在发展自己的一个开发工具，那就是MonoDevelop。MonoDevelop的目标之一，就是希望自己能成为一个真正跨平台的开发工具，而不仅仅只是开发出来的应用能够跨平台运行。从这一点来说，MonoDevelop对于跨平台的目标比Delphi Prism还要更进一步。而最新的MonoDevelop 2.2 Beta版本，也终于实现了这个目标。MonoDevelop 2.2是第一个Mono官方正式支持的可跨平台运行的版本，这意味着我们可以将原本在Windows操作系统中使用MonoDevelop开发的项目，复制到Linux或Mac操作系统中，然后在Linux/Mac中打开MonoDevelop，继续调试并运行我们的项目。

Java技术



■ 主持人：范凯
网络ID Robbin，JavaEye社区的创始人，开源软件的积极推动者和倡导人。

Oracle OpenWorld大会谈Java

本月，在Oracle OpenWorld大会上面，Java的创始人Jams Gosling专门谈到了Java的发展状况，他说：“在这个星球上，使用Java的设备比地球总人口还要多”。

当你谈论Java时，你自然会想起Java企业版，但是Gosling说Java EE只是冰山一角。Java的运行环境每周都会有1500万的下载量，而且在有新产品发布以及业务繁忙之时，下载量还会大大提升。

谈到手机，全球有26亿的移动设备中运行着Java程序。Gosling指出，除了苹果的iPhone，现在几乎买不到没有Java程序的手机了。Gosling还谈到了Java以下鲜为人知的方面：

- 是否认为Java运行速度缓慢？Gosling说：“在这十年间肯定是谬论，Java虚拟机HotSpot在性能上要比C/C++和Fortran要强。”
- 在使用多核处理器时你需要注意几个问题。目前主机处理器平均有四核，但到2030年，将增长至5220核。而Java EE架构在面对多核问题上要简单得多。
- GlassFish是世界上下载最多的应用服务器，每月的下载量都能达到100万。Gosling表示：GlassFish在这世界上已经泛滥成灾了，就像“烂片恐怖电影”一样。
- Java最具“魔力”的部分就是虚拟机，而不是编程语言。许多语言都可以在Java虚拟机上运行，如：Scala，Ruby on Rails，Groovy/Grails，Python，JavaScript等。

另外本月最值得关注的是：IntelliJ IDEA如今也终于决定开源，来和Eclipse和NetBeans更好地竞争。从即将发布的9.0版本开始，IntelliJ IDEA将为市场提供两种版本：免费和开源的社区版，完全功能的旗舰版（即从前的IntelliJ IDEA）。此举最重大的意义在于，社区版的引入，完全清除了那些挡在广大使用IntelliJ IDEA进行纯Java开发的用户面前的障碍——价格。这个版本不仅免费，而且——这也是最重要的——完全开源。IDEA开源带来的好处是能够给IDEA平台带来更多的第三方插件和开发者，更好地应对IDE市场的竞争。

软件工程·项目管理



■ 主持人：潘加宇
UMLChina 首席专家，潜心研究和实践 UML/UP 相关技术的应用。

方法需要理论

Visual Paradigm发布了其敏捷建模工具的新版本Agilian 2.1，其中一个新特性是提供了用例事件流编辑器。事件流编辑器支持多级事件步骤，扩展点以及测试案例计划，并且可以使用声音给模型元素作注解。

相对于面向对象建模工具，用例建模工具更像是二等公民，由于UML只规定了用例级别的表示法，路径步骤以及其他补充约束的表示法并没有作出规定，主流的建模工具对用例的建模只限于用例图，剩下的则由建模人员自行发挥，画活动图者有之，画序列图者有之，而大多数人还是用文本工具，如Word来书写用例文档。当然，也有一些厂商一直在坚持推出用例工具来改善用例的书写效率，例如CaseComplete 2009 R2，Visual Use Case 2009。

用例的提出者Ivar Jacobson和按契约设计的提出者Bertrand Meyer共同在DDJ上撰写了一篇文章“Methods Need Theory”（方法需要理论）。文章批评了软件工程界的现状，认为软件方法学本来应该是一门基于科学的工程学科，但现在有时看起来像时装行业，拣了丢，丢了拣，每季都有新款式；有时甚至像不良的政治，不同教派的领袖通过口号和宣传煽动人们的狂热，否认其他派别的价值。Ivar和Bertrand特别说到，现在流行的敏捷浪潮就生动地体现了上面的描述：把许多已有的方法和过程重新包装，宣传成新的；通过精心编写的宣言（Manifesto）来鼓动打破过去的旧方法。Ivar和Bertrand也提出了一个愿景：对方法学的本质作建模，发现其核心，即所有方法学之母，然后，使用这个核心来描述现在流行的各种方法学。现在已经投身到IBM Rational的Scott Ambler对此表示赞同。他撰写了文章“The Theory Needs s Strategy”（理论需要策略），认为愿景虽好，但实施起来难度很大。Scott为实现这个愿景提出了六条具体建议，还希望能够总结以往类似活动的经验和教训，例如20世纪90年代初模式社群为模式所做的工作、OMG为UML所做的工作以及Ivar Jacobson为Unified Process所做的工作。

Artisan推出了Windows 7下的建模工具新版本Artisan Studio 7.1，一个引人注目的新特性是glossUML，可以画出有3D视感的SysML/UML/UPDM（Unified Profile for DoDAF/MODAF）模型图。

Open Source



■ 主持人：叶亮
系统分析员，SCJP，Java 工程师，关注开源社区。

自信的Google和Oracle

Google在10月份发布了Chrome Frame，可让IE浏览器（包括IE 6、7、8）使用Chrome的WebKit核心以及高性能JavaScript解析引擎V8。Google声称，这款插件能立即改善IE浏览器的性能；同时还特别声明，Google认为不值得试图做出努力让Google Wave能够运行在IE浏览器上。“使用Chrome Frame，我们既可以把时间花在增加新功能上，又不会让IE用户受冷落”。微软和Mozilla都对此表示强烈反对，称此行为近乎“疯狂”，“会导致用户难以管理网站数据，影响上网体验，带来安全隐患”。微软特别强调，Chrome Frame会导致IE更容易受到攻击，用户现在需要同时担心IE和Chrome的漏洞；而且会导致IE浏览器的数项功能失效，包括隐私浏览模式。

Google在9月底发出首批10万个Google Wave邀请名额。作为一套即将诞生的产品，Wave引起了业界广泛的兴趣。它究竟能给大家带来什么改变？答案当然要等它对公众开放了才能清楚。不过Google的野心已经暴露无遗。试用了Google Wave的人表示，它就是Gmail、SNS、IM、Twitter和论坛的集合体。Google认为“电子邮件等传统互联网应用已经显示出疲态，肯定会被新的形式沟通方式所取代。”不过，要让众多的大型企业将邮件系统转移到Wave上并非易事，而Wave能否成功抢走Facebook、开心、MSN和AIM等的流量和用户时间，更要假以时日。

Oracle OpenWorld于10月11日在旧金山开幕。这是Sun和Oracle联姻以来，首次共同露面。Oracle宣布，在采用CMT的技术SPARC服务器和Solaris操作系统上部署的Oracle 11g创造了一项新的TPC-C基准测试世界纪录——12节点的Sun T5440服务器集群，7,717,510 tmpC，2.34美元/tpmC。结果证明，Oracle-Sun组合的运行速度超过了在IBM旗舰服务器Power 595上运行IBM DB2的速度。Oracle-Sun系统配置消耗的能量是IBM配置的1/4，而速度却快26%。Oracle-Sun基准测试比IBM基准测试的交易响应时间快16倍。

移动开发技术



■ 主持人：马宁
微软最有价值专家，Windows Mobile 开发者。

反击的号角

当世贸天阶巨大的屏幕上出现Windows Phone时，微软在中国的智能手机市场吹响了绝地反击的号角。Windows Phone君临中国的发布式，让人们重新认识了软件巨人的市场能力。这次发布式集合了七十多家OEM厂商，除了国际厂商外，中兴、海尔、天语等国内品牌也聚集在Windows Phone的大旗下。在即将到来的3G新时代中，寻找一个可靠的操作系统来对抗iPhone等国外厂商，已经成为国内厂商的当务之急。虽然Windows Mobile 6.5在功能上乏善可陈，但是在Android不成熟，iPhone、WebOS、Symbian不外卖的情况下，Windows Phone似乎成为国内厂商的唯一选择。而在经济危机的背景下，Windows Phone对于中国市场格外重视，所以本地化工作十分出色，不但有本地化的MyPhone服务可以在云端备份联系人、邮件等信息，而且也增加了MSN折酷等服务。在本身功能并不出色的情况下，合作伙伴与服务似乎成了Windows Phone的制胜利器。

Sun也开始了反击，准备推出Java的在线商店。Java商店里销售的大部分是Java ME程序，而不是Sun主推的Java FX，也有一部分是针对PC的Java SE程序。Sun也准备发布针对MID的JVM，在Java SE上增加移动特性。

而Android方面也有很多好消息，Gartner的报告说Android的市场占有率到2012年会超过iPhone和BlackBerry，从现在的1.6%上升到14.5%。同样授权给众多OEM厂商的Windows Mobile会从10.3%上升到12.8%。另外一个消息是摩托罗拉在开发出基于Android的CLIQ手机后，将正式退出LiMo基金会。这意味着摩托罗拉已经完成了Android的预研，准备全面发力了。摩托罗拉作为唯一一个在软硬件方面都具备强大研发实力的公司，对于Android阵营来说无疑是一个强援。

而Android的脚步也没有停留在手机和MID的市场，Hard Kernel推出了名为ODROID的掌上游戏机。Android内置了对OpenGLES的支持，而支持OpenGLES硬件加速的处理器也越来越多，所以开发掌上游戏机的时机已经成熟。SONY也开始推出PSP的在线商店，接下来的问题就是由谁去开发游戏引擎和在线商店。

数据库技术



■ 主持人：王翔
软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

Oracle Exadata 2更换硬件供应商 No SQL和非关系数据亮点频出

作为新一代甲骨文高性能数据产品，Exadata 2 Servers的硬件供应商变阵给业界不小的惊讶——甲骨文选择Sun作为供应商代替HP。在甲骨文总部一个面向互联网大众的展会上，CEO拉里·埃里森介绍了融合Oracle数据库和Sun FlashFire技术的新一代Exadata Server，新的Exadata Server不仅提供更高性能的I/O，同时也可以与云计算平台集成。不过业界对新产品能否真正大规模占领商务智能市场仍存在一些担心，毕竟这是一种相对新的产品模式，以往NCR虽然有类似的方案，但采用专用硬件系统，需要付出更多的成本。甲骨文宣布收购Sun之后，有关Sun硬件部分将被舍弃的猜想在业界传播甚广，HP和IBM也在极力争取这部分市场，但甲骨文的“数据库 + 硬件”捆绑政策凭借Exadata Server的高性能表现也许能成为市场意向的转折点，而且几周前甲骨文也明确表示将向Sparc和Solaris投入比Sun更多的钱，种种迹象暗示甲骨文不仅不会放弃Sun的硬件和系统软件，相反它会提高该领域与对手的竞争力，形成竞争优势逐次展开的数据库、企业应用软件、操作系统和硬件的多个战场。

近期数据库市场异常活跃，No SQL和非关系数据系统的消息频出抢走了主流关系数据库的风头。MongoDB作为高性能、无格式的开源数据库的代表，采用一种称为BSON的存储格式（二进制的JSON形式），它可以持久化类似诸如Map、结构体、数组和各种动态语言的数据对象。该产品采用定制的网络协议，据称可以获得比REST更好的性能，当前已经提供对Python、PHP、C++、Java、Perl的驱动，Erlang、Factor和C#的驱动也在孵化之中。

甲骨文发布了面向关系数据的Oracle Berkeley DB 4.5和面向XML数据的Oracle Berkeley DB XML 2.5；Sybase发布了一系列面向亚马逊EC2的支持软件；PostgreSQL在数据库管理方面也有进展，近期发布了SQL Manager 4.6版本。

互联网



■ 主持人：钱宏武
职脉网技术合作人，原搜狐互动产品开发部主管，资深互联网社区架构师，垂直搜索领域专家。

火热的手机操作系统

我用了很长时间的GDB，对于其多进程的调试一直很不满意，用一个词来形容就是惨不忍睹。在大家都快忘记上一个版本是何时推出的时候，这个月GNU终于推出了7.0版本，现在多核盛行，多进程的方式也成为业内最流行的编程模式，与BUG战斗多年的程序员们又多了一个新的武器，希望它别像GCC4一样中听不中用。

手机操作系统大概是这个月最为热门的话题，Google，微软都不断爆出新闻，就连移动也参与其中，微软推出自己的6.5，用过WM的人一般都会对上面企业推出的软件比较热衷，但在看过IPhone的App Store之后，才算是如梦初醒，原来应用的安装也可以这么简单清楚，界面可以如此好看。

本月正式登场的移动OPhone，也是我认为比较明智的举动。在9月初一个偶然的机会下，我曾试用过一下，确实有点惊喜。看来一流企业做平台在移动身上不是一句空话，而这个推出仅仅是个开始，如何赢得国内众多开发者的青睐，成为一个名副其实的平台，才是重中之重。

Google又是推出Chrome OS，又是搞Android，都是一副志在必得的样子，却爆出源码泄露、问题多多的消息，实在是祸不单行。

做手机系统，然后面向开发人员，推自己的App Store，使用者通过移动互联网直接下载使用，当足够多优秀的软件出来，这个手机系统的使用者会更多，相对的，将刺激更多的开发者。使用者开心，开发者赚钱，手机操作系统赚大钱，一个三赢的系统让大家皆大欢喜，没有污染，没有库存，移动互联网把一切都简单化。

而对于物联网来说，个人感觉它更像是一个政绩工程，政府在努力推动，又是国际标准，又是地方申请支持，但，现在国内最让人忧虑的依然是支付问题，其实解决了支付问题，真的上了物联网，收费才是最麻烦的问题。就大可以坐收渔翁之利了。但愿这个不是国庆60周年的一个献礼而已，毕竟我们太多的概念被抛弃过。

游戏开发



■ 主持人：赵青
九艺科技CEO，曾任金山西山居技术总监，网易方舟产品总监，《剑侠情缘2》项目负责人及主程序员，资深游戏开发者。

国内网络游戏杂谈

《魔兽世界》真是一款非常特别的游戏，它所引起的和游戏本身无关的事情远远超过其他产品。《魔兽世界》使得两家上市公司反目成仇。更换代理后，遥遥无期的审批，惹得百万玩家非常不满，更引出了文化部和新闻出版总署中国网络游戏监管之争。最近，代理《魔兽世界》的网易因为版署问题对相关政策的表述，导致连续3天股价跌去13%。

GDC（游戏开发者大会）在海外享有盛誉，但是自从其2007年以GDC CHINA的名称进入中国后，连续几届都不温不火。放下这个不说，从技术人员的角度看，国内自网游火爆之后，潜心研究技术的人越来越少。或者更加准确地说大家关注、研究的重点不同。GDC上关于单机或者视频游戏的主题比较多，基本没有与网游相关的讲座，即使有，我估计这些外来的和尚也念不好中国的经。单机游戏和网络游戏虽然都是游戏，但存在的差异性是显而易见的。大多数欧美开发的网游在大陆地区水土不服，已经可以说明，即使是那些需要我们仰视的欧美游戏开发大师，他们对网游的认识也不见得比我们更高。还有需要说明的是，由于盗版问题，中国目前单机游戏的开发人员基本已近绝迹了。

《摩尔庄园》是个很有意思的社区网游，细分用户非常准确——面向低年龄层。在此之前国外已有成功的同类产品。就目前的百度搜索排名来看，它已经非常成功。但由于网游的恶名，以及面向的用户是祖国的花朵，自然引起了很大的争议，央视两度炮轰《摩尔庄园》，小玩家爆吧激烈反击，只是一个开始，它所要面临的此类问题，还会越来越多。

巨人集团CEO史玉柱称，“游戏产品的成功只和游戏性能有关系。营销的作用，只是缩短这款游戏推广的时间。如果一个产品，可能吸纳50万人在线，如果营销搞得很差，可能要一年半的时间才能做到50万人；要是营销好的话，三个月就做到了”。巨人由于进入游戏行业时间短，缺少沉淀，但同时也缺少束缚，总会做出一些打破常规的事情，这对整个行业来说是个好事情。

安全



■ 主持人：肖新光
网名江海客，安天实验室首席技术架构师，研究方向为反病毒和计算机犯罪取证等。

角逐大战略

10月16日，美国总统奥巴马在白宫网站发布讲话视频，呼吁公众加强安全意识。令人惊讶的是，这完全不是泛泛而谈，奥巴马讲话中直接向美国公众提出了升级反病毒软件、不打开来历不明的邮件、交流等要了解对方身份、避免泄漏隐私和财务等非常具体的信息，这种对于公众个人安全细节的关注，释放出一种特定的信号。

而10月正是美国的“提高网络安全保护意识月”，从5月份的奥巴马国家安全报告到9月美国网军司令部的成立，美国正在表现出对网络安全前所未有的关注，这让我想起在我90年代刚刚进入这一领域时就听到美国前辈的那句名言“我们最大的失误是在过去20年没有用安全的意识去教育信息技术的第一批工作者，而他们却将去领导未来的20年。”如果掐指一算，这20+20年的时间即将过去，未来的美国信息社会将由一批具有安全意识的人领导，他们正在表现出前所未有的重视与投入，以及毫不掩饰的谋求一家独大的姿态和雄心。而欧洲、日本、韩国、俄罗斯等也莫不表现出一种急迫的跟进。

在提升整体的网络安全能力方面，中国也在做出一些自身的努力。10月11日，第三届CNCERT/CC网络安全应急服务支撑单位进行了改选，在全国数百家网络安全企业中，评选出了10家国家级、12家大区级和33家省级应急服务支撑单位。把更多安全企业的能力协调起来，共同抵御网络威胁的狂潮，这非常符合CERT体系本身的内涵，也创造了一道民间的产业体系防线来弥补国家的安全能力不足。

以往我们非常关注漏洞和攻防焦点，但在这个恢宏大气的十月，我们总能隐隐感到，不同国家战略与价值的全球角逐，一定会如无形之巨手，影响和改变着产业与企业、技术与商务、团队与个人的命运。

商务管理解决方案



■ 主持人：贾蓓
解决方案专家，曾获微软Dynamics CRM产品认证专家、微软CRM MVP称号。有多年企业信息规划管理、企业商务管理解决方案的项目经验。

甲骨文全球大会

对于甲骨文和Sun的联姻各界褒贬不一，但是甲骨文的CEO一直都信心满满，10月13日拉里•埃里森表示在收购Sun之后，甲骨文的实力将能同IBM抗衡。

同一天，甲骨文宣布收购美国媒体知识产权管理软件开发商Sophoi的所有资产，从而加强自己在媒体公司方面解决方案的力量，帮助企业把知识产权被侵权的风险降低到最低限度。

次日，甲骨文召开全球大会，会上公布了首届年度甲骨文合作网全球奖，用以表彰在技术软件、应用软件、行业软件、加速计划和中小企业解决方案等许多类别产品中的表现卓越的合作伙伴。并且在同时举行的第六届甲骨文北美Titan奖颁奖会上表彰了2009财年成功开发或交付的合作伙伴解决方案以及最佳实践。

大会上还发布了最新的商务智能软件7.9.6.1版。本版提供了与Oracle JD Edwards World预集成的Oracle支出分类(Oracle Spend Classification)的新组件，由此扩展了对第三方平台的支持。这个支出组件是原有商务智能软件中的新模块，它能帮助采购部门对支出明细进行总结分类，并且能分析企业购买模式的知识基础，采用针对机器学习和其他预测技术的Oracle数据挖掘来自动实现支出分类，从而便于采购经理了解整个企业的支出情况，以达到节约成本的目的。该新版软件还推出了一个新适配器，以同Oracle JD Edwards World财务管理（A9.2版本）集成。同时，该新版软件还提供扩展的本地化和平台支持。

此外发布的还有最新版企业人力资本管理产品套件：PeopleSoft Enterprise人力资本管理9.1、PeopleSoft Enterprise PeopleTools 8.50、PeopleSoft Enterprise学习管理9.1和PeopleSoft Enterprise门户9.1，以及针对黑莓的Oracle移动销售助理（Oracle Mobile Sales Assistant 2.0）和Oracle社交型CRM应用软件版本3，以加强现有的托管性CRM在移动和社交方面的功能。

现在各大软件公司都看重社交网络的潜在力量，纷纷同社交网络进行合作，或自行开发社交网络，以期借此扩大自己的影响范围。例如SAP今日宣布同商务社交网站LinkedIn合作，以通过在LinkedIn上对SAP的好评吸引更多的社区成员。同时PTC公司也宣布同微软合作开发社交型产品。

动态语言



■ 主持人：高昂

资源与环境信息系统国家重点实验室博士生，关注动态语言，喜爱写旅行游记，同时也是 OSGeo 中国和 InfoQ 中文站成员。Blog：http://www.gaoang.com/

云计算基础设施中的动态语言项目

云计算平台能够提供可动态访问的主机、网络存储等多种资源，目前 Amazon、Google、Joyent 等厂商推出的云计算服务已为开发者所熟知。与此同时，Eucalyptus、Enomaly、Nimbus、OpenNebula 等开源云计算平台的搭建方案也层出不穷。动态语言编写的代码不仅被部署在云端实现业务应用，在云基础设施的搭建环节，也出现了不少动态语言实现的云项目。

Vertebra 项目是 EngineYard 推出的新一代云计算平台，并在 GitHub 储存库上对开发者提供项目源代码。Vertebra 基于 Erlang 和 Ruby 构建，具备服务广播和发现机制的消息总线，用于编写分布式实时应用。Vertebra 可适应异构云计算平台的差异，进行自动化流程及应用的管理。其中，包含安全机制，服务发现机制，状态机工作流系统和分布式键-值存储系统在内的核心代码由 Erlang 编写，Ruby 语言被用来编写中代理的框架和实际代理。Vertebra 致力于实现安全、可拓展且易于管理的云计算组件，不仅能够运行在自有云基础设施中，也可部署在 Amazon EC2 或 VMware VCloud 等第三方云计算厂商搭建的环境中。

此外，开源项目 CloudCrowd 是一个完全使用 Ruby 语言实现的云方案，帮助 Ruby 开发者更为容易的编写分布式处理应用。CloudCrowd 设计目的是处理大规模且消耗资源较多的事务，比如完成大数据集迁移、视频编码、图片生成和处理、从 OCR 或 PDF 文件中进行文本抽取等工作。

相较而言，用于实现云存储框架中分布式 Key-Value（键值对）存储系统的 CouchDB 项目，有着更为广泛的用户且开发社区非常活跃。CouchDB 项目由 Erlang 语言实现，是 Apache Incubator 开发的分布式、容错、面向文档的数据库，目前最新版本为 0.9。CouchDB 底层是一个 B-tree 存储结构，与传统关系型数据库不同的是，其数据表仅包含文档 ID、文档版本号和内容字段。CouchDB 在应对网站并发性上具有很大优势，在基准测试中，CouchDB 可以应对多达 2 万个并发连接。CouchDB 对外提供了简便易用的 RESTful API 格式的接口，可以使用不同语言的客户端如 Python 等进行调用，并且 CouchDB 中数据的读写都可通过调用其 HTTP 请求实现。

虚拟化



■ 主持人：蒋海鸥

华中科技大学计算机系统结构硕士，计算系统虚拟化 973 项目组成员，系统分析师，SCJP，SCWCD，SCSA，SCNA

聚焦 VMworld 虚拟化盛会

IT 行业独立研究公司 TheInfoPro 最近的实时数据显示，2009 年安装的服务器中有 50% 将采用虚拟化技术，预计到 2012 年这个比例将增长到 80%。可见，虚拟化技术正在以较快的速度增长和普及。

万众瞩目的 VMworld 虚拟化大会于 8 月 31 日至 9 月 3 日在旧金山召开。作为虚拟化领域内的领导厂商，VMware 今年推出了长距离 VMotion 技术、VMware Go、PCoIP、CapacityIQ 与 ConfigControl，并开放云端平台与 vCenter 的 API 接口，强化第三方软件厂商的支持。其轻松部署虚拟机的 vCenter Lab Manager、可显示每个虚拟机账户信息的 vCenter Chargeback 等产品都让人眼前一亮。VMware 公司总裁兼首席执行官 Paul Maritz 在大会上强调了降低 IT 复杂性的重要性，对 VSphere 这款易于管理、并能很好地降低数据中心复杂性的虚拟化软件给出了很高的评价；VMware 首席技术官 Stephen Herrod 作了题为《虚拟化的未来：从手机到云》的演讲，提出了 VMware 的移动策略，除了将手机作为瘦客户之外，还可以通过手机管理数据中心，并对手机本身实施虚拟化。其 VMware View 产品除了可以作为便携终端的瘦客户机之外，还可以作为 vCenter 的管理终端使用。同时，他还表示 VMware 将公布实现手机虚拟化的“VMware MVP”(Mobile Virtualization Platform)。

与此同时，开源 hypervisor 之家 Xen.org 发布了 XCP(Xen Cloud Project)，旨在跨广泛的云厂商使虚拟化标准化，为 Xen 创建一个稳定的、定义好的 API，从而加速其在企业和云里的采用。软件巨头微软专为中端市场客户推出了一系列“特殊应用的虚拟化/管理软件以及服务升级”的解决方案，如 System Center Virtual Machine Manager 技术等。此外，思科与 VMware 合作，在大会上展示了一项远程实时迁移技术，整个技术可以让远在 125 英里外的虚拟服务器穿过多个数据中心，从而使企业真正撤离一个数据中心。

企业级开发



■ 主持人：胡长城
网名“银狐999”，普元软件R&D部门架构师。在Workflow、BPM、SOA领域有过多年的产品研发和实施经验。

企业信息化正在深入

IDC《2009年亚太区动态IT基准》调查的最新阶段性结果显示：业务流程和IT服务标准化，以及有助于使企业获得更多业务机动性的技术，仍居于最重要的位置。约有43%受访的中国CIO指出，他们2009年的企业IT技术更新战略将集中在服务导向型架构方面(SOA)；约有53%受访的中国CIO计划将主要应用软件转移到SOA环境下灵活的按需网络服务方面。

企业信息化正不断深入和升华，逐渐从早期的辅助性设施作用，转变为运营支撑作用，甚至是企业决策支撑核心体系。围绕这种转变，“流程（Process）”及“信息与数据”在管理层面的挖掘价值显得越来越重要。围绕这两个方面价值所提供的BPM和BI产品或解决方案，将成为企业信息化选择中的重中之重。

今年初就被IBM点燃的“智慧地球”火焰，伴随着SOA深入和云计算兴起，越来越旺。在IBM看来，智慧地球的路标已经清晰可见。新锐洞察（New Intelligence）、智慧运作（Smart Work）、动态架构（Dynamic Infrastructure）以及绿色未来（Green & Beyond）分别是四个组成部分。而在所有这些部分当中，实现新锐洞察（New Intelligence）的途径——商业智能（BI）是重中之重。

近日，国产BPM厂商炎黄盈动公司对外宣布，成功研发出中国首个专业AWS BPA业务流程分析平台组件，借助该平台可以帮助用户简单、方便地进行高层流程绩效分析，让管理者能够实时洞察企业的运营效率。

2009甲骨文全球大会(Oracle OpenWorld)已经在美国旧金山开幕（10月11日至15日），与往年备受关注的甲骨文的新产品和新策略相比，今年的Oracle OpenWorld一开始，人们关注的焦点就集中在了甲骨文收购Sun这件事情上。在甲骨文全球大会前夕，甲骨文CEO拉里•埃里森和Sun创始人兼董事长斯科特•麦克尼利向甲骨文和Sun的用户保证，当甲骨文斥资74亿美元收购Sun的交易完成后，甲骨文将继续投资Sun的大部分关键技术，从而提升两家公司的技术水平。

RIA技术



■ 主持人：徐基伟
徐基伟（花名崇厚），目前任职于淘宝网UED前端开发工程师，专注于JavaScript及其相关技术，业余也涉及桌面及移动应用的开发。

Canvas的前生、今世和未来

Web2.0大行其道，在日渐复杂的应用需求面前，传统的HTML、CSS、JavaScript等技术所能展现给用户的体验，有时候已显得捉襟见肘。此时，为了更好展现图形化具有丰富用户体验的Web未来，Canvas应重任而生。

Canvas最初在Mac OS X中出现，后被WHATWG接受成为HTML 5中的新元素。HTML 5规范离正式发布还有很长一段时间，各浏览器厂商已然按捺不住，纷纷宣布支持Canvas标签，目前Firefox、Chrome、Opera和Safari都已经支持Canvas。

通过各浏览器的广泛支持，加上JavaScript引擎的飞速发展，已给Canvas的普及打下了良好的基础。此时，Web开发人员可以开始考虑去学习和使用Canvas，用于改进自身产品的体验。在众多浏览器厂商推动下，微软仍逐渐感受到压力。随着IE的市场份额走低的趋势明显，微软在众多的压力面前，让IE支持Canvas只是时间上的问题。

对比Flash等RIA技术，Canvas的优势在于它是开放HTML 5标准之一，这意味着不需要额外去学习新的技术就可以使用Canvas开发出各种应用。目前Canvas在传统RIA面前略显稚嫩，但借助于HTML 5以及浏览器原生支持的优势，Canvas的未来不可小视。

Web应用与传统的桌面应用的界限会被逐渐的模糊，甚至在可以预计的将来，我们可以直接在浏览器中玩3D游戏。这势必会在目前略显龙钟的Web领域打入针强心剂，从而导致Web开发的再次变革。

在国内目前Canvas的应用才刚刚起步，对于Canvas的研究还处于起步阶段，但国际上已涌现了不少激动人心的应用。例如，使用JavaScript+Canvas实现的NES模拟器，YAHOO Pipes那样流畅高效的操作界面以及Google Chrome Experiments中那些酷炫的动画效果，如果使用传统的HTML+CSS+JavaScript来实现，这是完全无法想象的。搭着HTML 5的早班车，同时拥有众多浏览器厂商的支持，Canvas必然在不远的将来大行其道，让我们拭目以待！

2009中国软件开发2.0技术大会在北京盛大召开



10月22日，由CSDN和《程序员》杂志联合主办的2009中国软件开发2.0技术大会（SD2.0）在北京温都水城盛大开幕。开场致辞中，CSDN总裁蒋涛分享了今年SD大会的主题：占领技术制高点，实践分享开放协作。并着重探讨了该主题背后的平台变革背景。来自海内外的70多位知名讲师，通过三天、近百场的精彩课程，分享自己在专业领域内的实践经验和精湛观点。此次大会主要包含六大议题：新一代互联网、企业应用技术、软件工程管理、语言与工具、移动开发、软件创富。除精彩课程外，还安排了技术冲关、主题沙龙、欢乐Party等交流互动节目。SD 2.0大会是国内最大的由第三方媒体举办的软件开发大会，自2007年起，已成功举办两届，在开发者群体中有着极高的影响力。

中国软件行业协会成立25周年庆典 《程序员》荣获杰出传媒奖

9月19日，“2009中国（南海）软件和服务外包高峰论坛暨中国软件行业协会成立25周年庆典”在广东省佛山市南海区隆重举行。本次会议上，中国软件行业协会表彰了25年来推动中国软件产业发展的25位功勋人物、杰出人物、杰出企业家、脊梁企业、杰出传媒、品牌之星、杰出园区和行业信息化杰出供应商等。《程序员》杂志荣获“推动中国软件产业发展之杰出传媒奖”。

2009英特尔中国研究院开放日举行

10月12日，以“融慧中国，共创明天”为主题的2009英特尔中国研究院开放日在北京举行，会上展示了英特尔中国研究院近30项前瞻性技术研究项目和成果，包括企业级计算、终端技术、移动计算和学术合作创新四大技术展区。同期，英特尔中国研究院（Intel Labs China）以全新的名称和形象正式亮相，英特尔公司全球副总裁、首席技术官、高级院士兼英特尔研究院总监贾斯汀（Justin Rattner）强调，将致力于在中国打造世界水平的研究院。

拥抱2009自由软件日



2009年9月19日第六个国际“自由软件日”开幕。本次活动得到了自由软件基金会和中国开源软件推进联盟的大力支持，并由Sun、Mozilla、CSIP、中标软件、北京Linux用户组、龙芯俱乐部、哲思社区等多个企业和社区团体联合组建了筹备委员会。由遍布全国的20个省市、70多个分站活动由包括清华大学、北京大学等高校和众多开源社区共91家单位主办。活动遵循国际惯例，采用开放的组织形式，包括主题发言、技术讲座、现场演示、座谈、沙龙等。

首届“中科杯”全国软件设计大赛获奖名单揭晓

10月16日，由中国软件行业协会和中国科学院软件研究所联合主办的首届“中科杯”全国软件设计大赛在北京落下帷幕。本次大赛于今年4月启动，分别以“展现软件技术的科普作品设计与制作”、“移动互联网设备游戏设计”、“基于开源软件的应用软件架构设计与实践”为主题，吸引了来自全国200多所高校、研究所的本科、硕士和博士参与。大赛中，北京理工大学团队的作品“PuppyFish”获大赛特等奖，大连理工大学和解放军理工大学团队则分别获得一等奖。颁奖式后，中科院软件所启动了“软件技术博物馆”的正式揭牌仪式，这将是依托中科院软件所建设的我国首家以计算机软件为主题的软件技术博物馆。

微软全球发布Windows 7三大亮点受关注

10月23日，微软新一代操作系统Windows 7在全球统一发布。作为微软全球研发集团的一支重要力量，微软中国研发集团在Win7开发过程中充分体现了中国“智”造的力量。这一全新的操作系统，有三大亮点值得关注。亮点一：最具中国特色；亮点二：“有病”自动诊断；亮点三：硬件兼容性强。

msup®
致力于培养企业级中高端研发管理人才
www.msup.com.cn
Service Hotline：400-812-8020

俄罗斯软件业面面观

编者按：《程序员》创刊伊始，曾开设“走向海外”专栏，报道国外软件业发展现状及不同特色，文章广受好评。九年之后，这些国家的软件业发展如何？带着疑问，本刊邀请俄罗斯软件外包领域知名专家、Auriga公司总经理Andrei Pronin先生撰文，为我们揭开当今俄罗斯软件业的真实面纱。

■ 文 / Andrei Pronin 译 / 卢鹄翔

对于我这个已经在软件业浸淫十数年，并且掌管着俄罗斯最知名软件研发外包公司之一的人来说，俄罗斯软件行业的方方面面，成败得失，看起来都是如此的明显和自然。然而，当我得到机会，能够与《程序员》杂志的读者朋友们分享个中滋味时，却意识到这绝非易事。在本文中，我将竭尽所能阐述一些基本问题，包括何为俄罗斯软件业，它如何开端，有着怎样的内部结构，与世界其他地区有何不同，在全球软件开发的进程中又是处在什么样的地位。

开辟鸿蒙：从高等科研院所起步

第一手落子，往往便决定了一盘对弈将如何开局。俄罗斯软件的传统很早（远在苏联时期）便伴随着科学密集型任务一同起步。苏联的第一个计算中心1955年在俄罗斯最大的高等学府（国立莫斯科罗蒙诺索夫大学）建立，此后的很长一段时间里它都是全俄领先的计算研究机构。最初的软件系统是针对诸如气象学、无人和载人航天计划、空气动力学、电动力学、结构分析、数理经济学等领域开发的。在数值分析和规划理论方面都取得了重大突破。此后，特别的“科学城”在全俄不同地区建立起来（例如莫斯科附近的泽廖诺格勒，或者新西伯利亚附近的“学术城”阿卡杰姆戈罗多克）；还有精心筹划的科技学校，它们成为了当时最先进技术与思想的前哨站。

当苏联不复存在，IT舞台的领导角色便由第一批出现的商业软件公司所继承。然而，这些公司通常都具有科学领域的渊源。一个很好的直接继承例子就是我所管理的公司——Auriga，它是第一家为来自美国和西欧客户提供软件研发服务的私营企业。1990年Alexis Sukharev创办了这家公司，他是国立莫斯科大学教授，这所俄罗斯著名学府的许多毕业生也汇集于此，为客户开发新的软件产品。正是由于对俄罗斯IT服务产业发展的贡献，Sukharev博士被《纽约时报》称为“俄罗斯外包教父”。

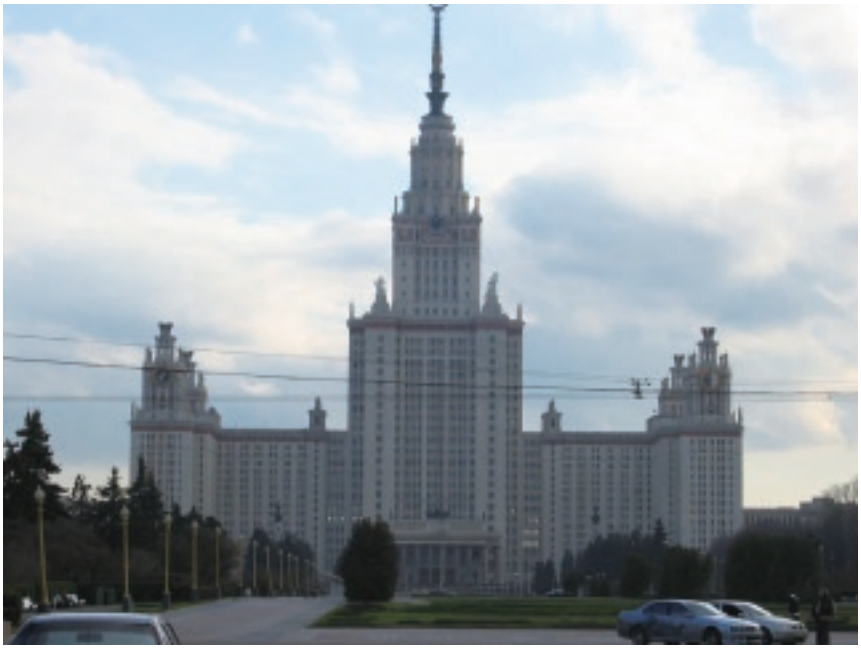


图1 世界数学界的最高殿堂——莫斯科大学

再晚一些（20世纪90年代初），当多种新兴产品出现，多家年轻的IT服务企业成立后，俄罗斯IT和软件产业便走上了集约型发展的道路。许多在那个时期创办的企业，现在已为广大计算机软件用户所熟知。仅举几例，如ABBYY、卡巴斯基实验室、Paragon Software都是90年代初期创立的俄罗斯品牌。

俄罗斯软件业的第二部分（软件服务）有它自身的特点，与这一领域其他典型，诸如印度、中国在许多方面有所不同。首先需要指出的是，由于起源不同，当前提供IT软件相关服务的俄罗斯公司，主要被分为两个大类。

第一类由一批系统集成商组成。最初，它们都是专注于国内市场的IT硬件经销商，经过多年经营，直到建立起稳定的硬件销售渠道后，它们逐渐开始提供IT服务，并为其客户提供某方面的软件开发。即使经历了21世纪研发外包浪潮的洗礼，这些IT企业也只是依靠其子公司（通常以不同的品牌运作，与其母公司并不关联）拓

展业务，而其自身仍着眼于系统集成、咨询和硬件销售。

第二类企业，它们一开始就将业务方向瞄准在为其西方客户提供软件开发服务上。这部分公司还包括了一些20世纪90年代的“老前辈”，诸如Auriga、EPAM Systems、Mera Networks、Reksoft、Starsoft、Development Labs（目前已被Exigen Services并购）、VDI（目前已被EPAM并购）等。这些企业将主要精力放在了美国、西欧、德国（包括德语国家）以及斯堪的纳维亚半岛。

不足为奇，到20世纪90年代末，这些处于领导地位的企业，便在迅速成熟的软件社区中发展成为行业协会的中坚力量。其中一个例子就是RUSSOFT协会，它由处于软件开发领域主导地位的多家公司组成，成员来自俄罗斯和前苏联的一些共和国。该协会最初称为“Fort-Ross”财团，在1999年9月9日成立，地点位于圣彼得堡，在同国家软件发展协会（NSDA）合并后，2004年9月更名为RUSSOFT。目前，该协会由80多家公司组成，拥有超过7000名高素质的软件工程师，代表着俄罗斯IT外包产业的形象。

长风破浪：高速发展的软件产业

俄罗斯软件产业是随着国家整体经济一同前进的。在全球金融危机之前，俄罗斯保持了连续9年的经济增长，年均GDP增幅为7%，2007年达到了7.7%。同期IT产业的增幅远远高出全国平均水平20%，达到27%，2007年软件产业增幅更是高达63%。就在软件产业变得日渐重要时，却在2008年遭遇到了金融风暴的严重冲击。然而，这不过是征途中的短暂停歇，软件业仍将朝着成为俄罗斯最重要经济产业之一的目标向前迈进。

目前，俄罗斯政府高度重视促进产业多样化，发展有竞争力的非资源产业，以及培育知识型或者“创新型”经济。为配合这一理念，政府出台了一系列措施和政策，期望能够提升高科技产业的竞争力，刺激创新型经济的发展。这一系列措施主要依靠政府有选择地进行干预，促进特定产业和区域经济的发展，包括设立经济特区、建立科技园区、国家风险基金、国家开发银行、税收优惠、培训计划、鼓励出口以及政府直接参与某些经济领域。

俄罗斯前总统普京和现任总统梅德韦杰夫都将建立新的知识型经济体系放在了首要位置。得益于超过6000亿美元的国家财政支持，许多新项目在近些年得以上马。普京总统2005年倡议的高科技园区就是一个很好的例子。Federal Task Force项目于2006年确立，2007年开始兴建，并将在2010年竣工。11亿美元政府投资和估计超过70亿美元的私人投资将用于该项目的建设。每个高科技园区占地40至100公顷，包含10万至20万平方米办公楼，以及25万至35万平方米的住宅、生产设施、社会基础设施、教育或研发中心。对将俄罗斯打造成为高端知识型项目的外包国来说，这些项目的成功，是最有力的推进手段。

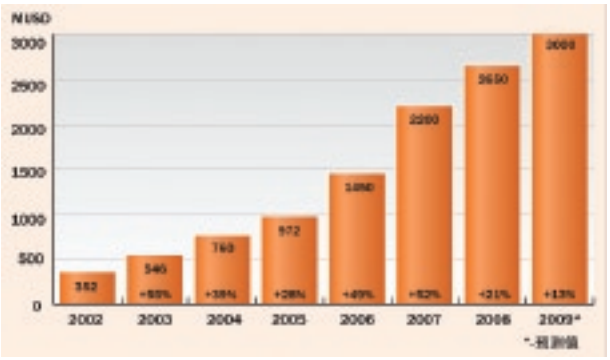


图2 2002-2009年软件和服务出口，以百万美元（MUSD）计算

俄罗斯成为IT强国的主要障碍，在于对IT行业缺乏相关的税收鼓励政策。虽然有过不少引入新法规的尝试，但事实证明其中大部分收效甚微，有的存在缺陷，或者根本起不到任何作用。主要问题是所谓的统一社会税（Unified Social Tax，这一税种依据员工薪资水平向雇主征收一定比例的税款），IT行业与石油行业的税率相当。然而在石油行业中，员工工资只占生产成本的一小部分，在IT行业，这个比例则超过50%。鉴于近期通过国家杜马初审的新法案，能够为软件公司带来额外优惠，我们预计这一情况将在2010年有所改观。

三足鼎立：服务、产品出口与外包

在俄罗斯软件产业结构中，服务和产品出口扮演着重要角色。由于我的整个职业生涯几乎都致力于软件研发外包，下文中，我将对其予以更多关注。

从2002年到现在，俄罗斯软件产品出口额增长了将近10倍，2008年达到26.5亿美元，约占俄罗斯IT产业出口总额的50%。不幸的是，由于全球金融危机，2008年软件出口增幅仅为21%，相比之下，2002年至2007年的复合年增长率则超过44%。尽管如此，整个行业仍在继续前行，即使在艰难的2009年仍将有所发展，预计到2010年，增长速度将恢复到金融危机之前的水平。

当然，距离成为全球软件市场的领导者，俄罗斯还有很长一段路要走。2009年全球软件市场总额预计约为2226亿美元（Gartner数据）至3880亿美元（Forrester Research数据）。即使包含国内和出口市场，俄罗斯所占份额也仅为1.5%~2.5%。假如依照软件开发企业的实力以及软件开发人员的专业水平，俄罗斯本应占领更多的国际市场。为了洞悉制约俄罗斯软件产业获得更大市场份额的细节，让我们一同探索软件出口中更深层次的结构。

俄罗斯软件出口基于三大支柱。

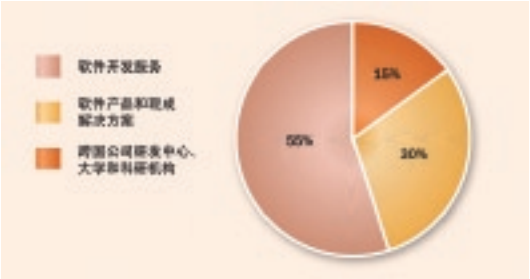


图3 软件出口份额

其中最小的（约占软件出口总量的15%）是跨国公司研发中心和科学研究机构。目前已有十多家大型跨国公司在俄罗斯设立了自己的软件研发中心。其中包括阿尔卡特、Borland、克莱斯勒、戴尔、爱立信、Google、惠普、华为、英特尔、LG Softlab、摩托罗拉、三星、西门子、Sun Microsystems、Teleca、T-Systems公司等等。“俄罗斯联邦提供了良好的经济环境，训练有素的技术人员以及出色的教育机构。”——IBM全球开发副总裁Rod Adkins如是说。惠普公司执行副总裁，CSO兼CTO Shane Robison也有相同的观点：“俄罗斯飞快的发展速度，深厚的技术底蕴，使其成为惠普公司拓展全球研发能力的绝佳据点。”

最近多份调查显示，尽管受到全球经济问题影响，俄罗斯IT业的投资前景仍然被普遍看好。根据各大报章和期刊，半数以上的世界大型公司都与俄罗斯存在软件外包业务联系。最近9个月中，微软、思科（与UFG联盟）、Sun Microsystems和其他一些公司已经相继宣布了在俄投资软件开发产业的计划。拟议的投资金额从1000万美元到3亿美元不等。

产品供应商是俄罗斯软件出口的第二大支柱，占出口总额的30%。其中一些产品颇具实力，包括Parallels的虚拟软件，ABBYY的电子词典和文字识别产品，卡巴斯基实验室的反病毒产品，Speech Technology Center的语音识别软件，Transas的船舶控制及导航系统等。即使从这其中我们也可以看出，俄罗斯的软件产品已经在多个领域取得举世公认，这些领域通常都需要具有数学、自然科学以及算法优化方面的相关背景。

软件服务提供商对俄罗斯的软件出口贡献超过50%，是其最大支柱。它们在全球市场上也相当成功，从英特尔俄罗斯区总裁Steve Chase的话语中，我们就能看到这些成功背后的缩影“如果您有棘手的任务，请交给美国人；如果您有困难的任务，请交给印度人；如果您有不可能完成的任务，那么，请交给俄国人吧。”纵观俄罗斯整个软件业历史，复杂计算和科学密集型软件已经成为传统优势项目，作为这一趋势的延续，高科技客户（需要为尖端技术提供相应的软件产品），便成为许多服务提供商的最爱和工作战略重点。

钟灵毓秀：优秀而创新的人才资源

早在2003年，Aberdeen Group就研究发现，俄罗斯IT服务提供商的客户中，67%与其同处于IT产业，其中41%为独立软件开发商。根据此项研究，俄罗斯的“工程师和软件开发人员最常参与的活动包括：核心应用程序开发项目、项目规划、需求收集与分析和规范设计。尤其需要指出的是，这些活动通常与软件离岸外包无关”。用一句话来说，如果您拥有一种高科技产品，并考虑将一些开发任务外包，那么，最好的选择就是交给俄罗斯。

为了理解这一现象，人们应该谨记，俄罗斯的科技潜能深深根植于其世界级的研发传统，此外，还拥有一大批科学家、数学家和其他IT专家作为后盾。对于从事复杂的研发项目，这些人具有“遗传”般的记忆力，此外，他们还从苏联时代的教育体系中继承了非凡的工程技术创新能力。俄罗斯的竞争优势在于其高品质的教育和训练，使学生专注于深层次的基础知识。一旦情况需要，IT毕业生便能够快速地学习新技术，适应技术变革，而不是仅仅局限于当前的热门技能。

为了在产品研发能力方面取得领先地位，发展高水平教育，目前已经成为俄罗斯的首要任务之一。全世界科技专家得出结论“尽管俄罗斯有着高水准的科学教育”，仍然有必要“在某些时候像传统教育方式那样，把尖端技术与理论知识结合起来”。教育是经济发展和社会繁荣的基础，提高俄罗斯的IT教育质量，与国际接轨，改善教学，像装备俄军那样以科技装备教育，这些工作任重而道远。

尽管困难重重，俄罗斯仍然是一流工程专家和技术能手的摇篮，它所培育的人才有能力并且乐于面对各种复杂难题。这一结论从ACM国际大学生程序设计大赛（ACM-ICPC）的结果中得到了印证，ACM-ICPC是国际顶尖大学团队间的程序设计竞赛，俄罗斯也曾自豪地合作主办过该项赛事。

2009年再次成为俄罗斯大学生丰收的一年，圣彼得堡精密机械和光学大学成功卫冕，并获得了该校的第三个世界冠军头衔。在这项赛事中，俄罗斯选手总共获得4枚金牌及1枚银牌。

今年的比赛结果也延续了以往多年中俄罗斯院校的获奖趋势。例如，在2008年，俄罗斯团队将决赛12枚奖牌中的5枚（2金、1银、2铜）收入囊中，2007年获得4枚奖牌。从2000年至2007年，俄罗斯总共获得了4次世界冠军。在过去3年的奖牌榜中，我们能够发现8支来自俄罗斯不同地区的团队，分别是莫斯科、圣彼得堡、萨拉托夫、伊热夫斯克、彼得罗扎沃茨克、阿尔泰和新西伯利亚。

我很高兴看到中俄两国在这方面携手并进。该项赛事的桂冠多为俄中两国选手所包揽，与中国选手间的竞争也在逐年加剧。

在其他全球性程序设计竞赛中，俄罗斯程序员也多有斩获。值得注意的是创新杯（Imagine Cup，由微软主办），

这项竞赛鼓励青年人发挥他们的想象力、热情和创造力，为技术带来革新，让世界从此变得不同。2009年，俄罗斯团队在竞争最为激烈的项目——软件设计中取得第二名；2008年，俄罗斯团队获得了杰出工程成就奖，并在Project Hoshimi竞赛中拔得头筹。过去几年中，多支俄罗斯团队在创新杯软件设计邀请赛中问鼎。俄罗斯程序员取得高排名的另一个例子，是在Google编程挑战赛（Code Jam）上取得冠军和季军，这项竞赛中，学生和专业程序员要在规定时间内解答复杂的算法难题。

优秀学生渊源不断，使俄罗斯劳动力市场拥有巨大潜力。根据Frost & Sullivan统计，每千人中软件开发人员的数量俄罗斯排名世界第一，科学家和工程师数量排名第三，遥遥领先印度和中国。根据联合国教科文组织（UNESCO）和德国联邦统计局数据，俄罗斯拥有世界上数量最多的科学专业学生。

尽管如此，人才数量不足仍然是俄罗斯将要面临的主要挑战。2007年，俄罗斯IT劳动力市场供需数量分别为100万和150万。经过金融危机，劳动力资源市场的供需紧张程度已经得到大幅缓解。如同许多CEO提到的，这场危机平衡了劳动力资源市场，目前已经变为以需求为导向，而非由供应主导。但是，有理由相信，一旦IT产业从这场危机中恢复，尤其当俄罗斯开始占领更大的国际市场份额时，人才短缺问题还会卷土重来。

继往开来：未来的俄罗斯软件业

让我们回到关于国际市场的论题上来，俄罗斯在软件研发和软件产品开发领域颇具天赋，尤其是高科技产品，需要依靠坚实的基础学术研究。

得益于此，俄罗斯已经具备发展成为全球软件研发中心的有利条件。英特尔数字企业集团（DEG）总裁Tom Kilroy在海得拉巴的一次会议中，发表了题为“英特尔创新：中国、俄罗斯和印度”的演讲，他强调，俄罗斯的“软件开发创新”与中国的“营销创新”，印度的“卓越芯片设计”同等重要。他这样描述俄罗斯工程师：“他们是杰出的软件开发人员，具备非凡的问题解决能力，基本功（数学和科学）极为扎实，他们有征服难题的欲望，而不只是研究研究规格。”他还称赞俄罗斯工程师“在业界无与伦比”，这也是业界的共识。

有人觉得，俄罗斯软件的发展方向，使其在同印度等全球IT外包强国的竞争中，不能大展拳脚。而我却认为，这是一股自然凝聚的力量，俄罗斯未来的软件事业可凭借这股力量蒸蒸日上，问题的关键在于如何正确地使用它。

首先，显然需要使用更积极和更系统的方式发掘这一产业蕴藏的巨大潜力。一直以来，我们都对外包产品研发抱有很大兴趣。但实际上，俄罗斯作为绝佳的外包项目提供国并未被所有的潜在客户所了解。例如，Zinnov管理咨询公司最近公布了一份“全球研发服务提供商”20强名单，只有Auriga一家俄国公司位列其中。看到公司20年的拼搏奋斗得到承认当然让我感到自豪，但我明白，仅有一家公司在列，远非俄罗斯企业能力的真实体现。

另外一项重要措施是发掘俄罗斯工业中的次优势项目。2007年，IDC公司在一份报告中强调，某些“软技能”使俄罗斯有别于其他国家，使之成为绝佳的离岸外包业务提供地。这些“软优势”包括同美国和欧盟客户间将近的文化、类似的工作理念、对商业问题的清晰理解以及相对较少的语言障碍。在Auriga公司的漫漫的成长道路上，我们很早就意识到了这些“软技能”的重要性，并将其作为公司战略重点进行培养。如今，随着整个俄罗斯工业的快速成熟，越来越多的公司开始将这一自然优势转变成为胜势。服务提供商开始寻找技术之外的着眼点，将精力集中于任务签订与交付的灵活性、完整性、文化接近程度之上，将工作重点转向对客户商业需求的关注，而非盲目遵从规范。如果运用恰当，这些“软技能”有可能成为俄罗斯软件服务业的制胜法宝。

世界著名的科学权威Esther Dyson在给《纽约时报》的文中写道：“俄国程序员不是单纯的程序员，他们是以软件开发谋生的数学家和科学家。他们长于解决复杂的大型技术难题。他们不是简单地随波逐流，他们最大的特点就是创造性，因此他们可以放心地把他们的方案付诸实践。”

作者简介



Andrei Pronin，Auriga公司总经理，负责公司的战略规划和日常运营。莫斯科国立大学理论力学和应用数学博士，在软件行业工作超过15年，10年专注于软件研发外包。1998年加盟俄罗斯软件外包龙头企业之一Auriga，历任技术总监、业务发展总监等多个工程和管理职位。在俄罗斯外包行业中，Andrei具有很高知名度，曾代表Auriga和俄罗斯参加过多项国际性商业活动，包括2009年6月Gartner外包峰会及CeBIT'07等。

南国热风 广东特色

——走进广东软件科学园



■ 记者 / 韩磊

建设者说

从位于广州市中心的广东省政府出发，向东偏北方向行进23公里，进入幽静的广州科学城彩频路东段。在这条不太宽的路上，最引人注目的建筑群，就是广东软件科学园了。

广东软件科学园筹建于2000年8月，是由广东省人民政府投资、省科技厅组织筹建、省生产力促进中心管理的软件产业公共技术资源服务机构和软件企业孵化创新服务机构，属广东省重大建设项目。

改革开放后，以“三来一补”为特色的制造业在广东省迅猛发展，但信息化程度却亟待提高，软件科学明显滞后于经济发展。形势逼人，广东省政府当即拨款1亿元人民币，在广东省科技厅编制下成立了广东软件科学园筹建办，时任广东省科技厅办公室副主任的经济管理博士文俊伟受命筹建广东软件科学园。

软件园建设宗旨是通过自身良好的技术资源、配套设施和投资环境，协同国内外软件产业界现有资源，从行业协调、引导着手，为业界提供技术/产品研发、评测认证、产品项目孵化、出口企业成长培育提供良好的行业发展环境及公共技术资源支持和服务。政府财政做首期支持，国家和地方共建，公司化运作，这是省有关部门给软件园定的调子。在筹建工作后期，广东拓思软件科学园有限公司正式成立，具体负责软件园的经营。

文俊伟现在拥有广东省生产力促进协会副主任的和拓思公司总经理的双重身份，这正好和软件科学园亦“官”亦“商”的体制相对应。“人家感觉你是政府，事实上我们是企业。如何在政府行为和企业的效益及其可持续性发展之间找到平衡点，这是比较困难的事情。”回顾9年来的发展历程，文俊伟感慨良多。

政府初期投资的1亿元说少不少，说多不多。2000万元买下100亩地，4000万元建起一幢综合楼，还有4000万用来建设数据中心，三个项目下来，钱就没了。9年后的今天，园区内共有七幢办公大楼，建筑面积约14.8万平方米。这七幢大楼的建设资金，基本上是由拓思公司运营所得再投资。仅靠物业和公共平台租用等收入，根本无法支撑这样大的资金投入。

文俊伟以潮州人天生的精明、博士的学养和政府服务的经验，和他的经营班子一起，创造了一种特殊的滚动发展模式。在为软件企业提供公共服务的同时，拓思通过控股、参股等形式，布局自己的IT产业格局。迄今为止，拓思控股、参股的企业已达九家，其中有两家是广州市重点企业。拥有



广东软件科学园筹建于2000年8月，是广东省软件共性技术“产、学、研一体化”研发基地。与天河软件园、南沙资讯科技园三园组成“广州软件园”，为国家软件产业基地之一。园区内设有国家863软件专业孵化器、广东省研发中心、广东省重点实验室等专业软件基地。广东拓思软件科学园有限公司负责该园区的经营和管理，实行企业化运作。

软件园投资的半官方背景，显然对这些企业的经营有所帮助。但在广东这个百年商埠和昔年的改革开放最前沿，光靠背景、关系是行不通的。拓思控股、参股的企业，在各自的专业领域均具有相当实力，其中不乏承建亚运会公共信息平台等大工程的广州乐庚信息科技有限公司这样的优秀企业。

回归公共服务

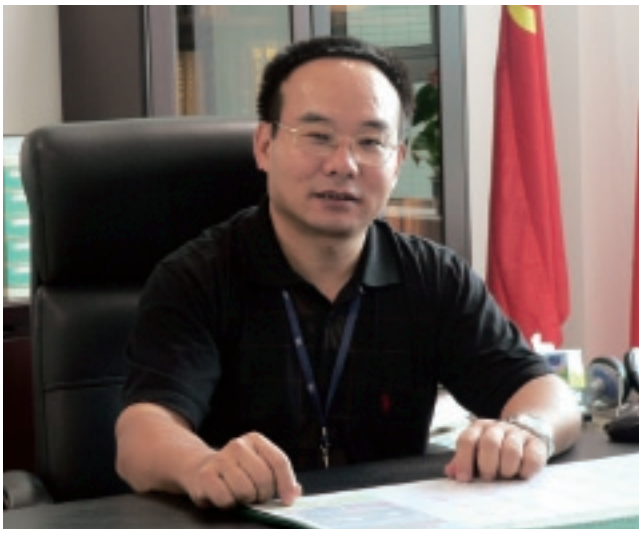
作为政府的“代理人”，拓思公司的本职是建设园区公共平台、为入驻的企业服务。拓思公司综合发展部副总经理荆晔说：“广东软件企业普遍规模小、分布散，要他们自行采购各种平台、工具并不现实。广东软件科学园从一开始就注重搭建公共支撑体系，为软件企业提供平台、工具租用服务。”

综合楼是园区内建成的第一幢办公楼。广东省软件共性技术公共实验室、网络数据中心、广州IBM软件创新中心、广东省嵌入式软件公共技术中心、广东省数字家庭技术支持中心、广东省中小企业信息化创新服务中心等技术资源服务机构分布于各个楼层。

2005年建成的广东省软件共性技术公共实验室，为广东省重点电子行业产业技术进步提供支撑服务。公共实验室采用“开放、联合、共享”的理念以及“边建设、边研究、边开发”的思路，为企业提供资源共享、产品技术研发、技术咨询等服务。截至2008年底，实验室共承担各类科技计划项目45个，包含国家级项目10个、省级项目28个。仅2008年，公共实验室就新立项目 18个（其中国家级项目3个、省级项目8个、市区级项目7个），完成了4个项目验收，国家863计划、国家支撑计划项目、粤港关键领域重点招标项目等6个项目的执行检查。

目前，公共实验室已建成包括嵌入式软件系统技术实验室、软件工程技术实验室、数字媒体技术实验室、家电行业共性技术实验室、移动终端行业共性技术实验室、共享软件国际化共性技术实验室在内的6个分实验室，并已在东莞、汕头、佛山、惠州、中山等地建立了服务网点。通过科技成果转化和技术服务，累计为广州及珠三角地区的6000多家企业提供技术服务和咨询。2008年，实验室公共技术支撑体系为园区企业、广州天河软件园、广州科学城以及珠海、深圳、中山等珠三角周边地区350多家电子信息企业的技术产品研发、质量保证和软件产品的运营服务提供了技术平台和资源支持。

广州IBM软件创新服务中心是软件园与平台级厂商合作的尝试。创新



广东拓思软件科学园有限公司总经理文俊伟博士

中心根据本地企业客户需求，结合IBM Rational技术，为企业提供软件工程、质量保证等方面的技术咨询、支持和解决方案。已有客户包括安利（中国）有限公司、中国银行深圳分行、宝钢集团、广州海格通信集团、远光软件、广州市信息中心、上海市浦东新区人民政府等企业和政府。

另一个与企业合作的尝试是广东省嵌入式软件公共技术中心。嵌入式软件公共技术中心的合作对象是广东美的集团，主要面向信息家电、汽车电子、数控机床、消费电子、网络通讯等嵌入式新技术、新产品领域，是广东省信息产业四大研发中心之一。美的与广东省政府共同注资建设嵌入式软件技术中心，无疑是认识到了科学技术在企业未来发展中的决定性作用。放大到珠三角地区乃至广东全境，对企业信息化的重视，已上升到前所未有的高度。

文俊伟认为，为软件园持续发展积累资金的任务已经基本完成。“随着时间的推移和投资回报的获得，（拓思）公司接下来要转向纯粹为软件企业提供公共服务的方向，更多的发展机会留给子公司和参股公司。”此举将拓思从身份困境中挣脱出来，回归其追求社会效益的本职。独立发展的原参股、控股公司，也将和软件园乃至全省、全国其他软件企业站到公平的竞争位置。

聚焦嵌入式软件与集成电路

截至2008年底，广东软件科学园在园企业230多家，园区企业总注册资本约20亿元，年产值约58亿元，上缴税金约4.5亿元，总就业人数超过2300人。从开园至今，广东软件科学园园区产值累计约220亿元，园区企业上缴税金累计约15.7亿元。

拓思控股、参股的企业，除负责园区物业管理的锐启公司外，已在各自的领域成为园区内企业典范，这也反映了园区在入园企业优选方面的倾向性。

软件园于2008年1月与海格通信集团联合成立的广州润芯信息技术有限公司，重点开发支持中国自主的北斗卫星导航定位系统的射频核心芯片。2008年，润芯公司在面向卫星导航定位和无线移动通信两个应用领域的射频芯片上，组织申报各级政府科技计划项目7项，其中国家级项目3项、省级项目3项。润芯公司代表了园区对集成电路设计和嵌入式软件开发企业的重视；专注于楼宇智能化设备技术研发、生产和销售的美的自控，在美的集团多元化产业战略中占有重要地位；从事软交换和多网融合终端的汇思通讯，成功拿下了广东省数字技术支持中心运营管理项目。

珠三角是中国最大的集成电路消费市场，在无线领域移动通信、卫星导航终端核心芯片和消费电子领域音视频芯片等方面有广阔发展前景，具有发展行业集成电路的良好市场需求基础及与整机互动发展的独特产业环境。广州鸿芯微电子致力于SoC集成电路设计开发，是中国闪联标准工作组会员单位。东莞泰斗微电子、东针电子（广州）也是行业中不可忽视的企业。

软件出口路漫漫

软件外包是园区重点推动的行业发展方向，也是省科技厅、工信厅、外经贸厅等政府部门一直在倡导的工作。2006年，科学园联合多家单位发起成立广东软件出口委员会，并被授予“广东软件出口基地”称号。政府表态并不意味着产业一定能兴旺。目前，广东软件外包产业还处于并不成熟的阶段。就广东软件科学园内部来看：一部分外包企业面向欧美，企业规模都不太大，七、八十人团队较多。此外还有两、三个对日软件外包企业。文俊伟认为，“经过几年推动，政府和企业都有发展软件外包的意愿，但去年的金融危机阻碍了这个势头的发展。”他相信，“危机过后，在可预见的将来，也许就是明、后年，软件外包会是广州软件企业转型的方向。”

在大概还算不上产业的广东软件外包产业中，广东华智和大宇宙（中国）广州分公司是较有盈利能力和代表性的企业。华智拥有300多人的开发队伍，是华南地区规模最大的对日软件外包企业。大宇宙是日本上市公司Transcosmos的分支机构，有着鲜明的日本软件企业特色。

共享软件也是软件出口战略的重要组成部分。拓思旗下控股企业广州神划信息技术有限公司从2005年开始与著名的全球软件在线行销渠道Digital River接触，在共享软件营销、项目孵化、人才培养等方面展开合作。共享软件行业进入门槛较低，但营销

是门高深的学问。神划在与DR的合作过程中，逐渐了解和掌握了共享软件制造与行销的有效手段。谈起共享软件，神划公司总经理江南对其又爱又恨。爱的是这个产业巨大的市场与前景，恨的是“有些年轻人，在我们这里学会了怎么卖软件，就离开公司自己去做。”除了行业进入门槛较低外，国资背景导致的人才管理与培养机制不够灵活也是问题。“在我这里一个月几千块，出去自己做共享软件，几个月后收入就远高于此。”江南认为，如果拥有更灵活的回报策略和职业规划措施，情形应该会不一样。

超越软件：产品质量认证

拓思综合发展部副总荆晔可能是最熟悉园区及园内企业情况的人之一。在采访中，她一再强调园区名称中的“科学”二字，并以产品质量认证业务为例，说明这两个字代表着园区不仅限于支持软件产业，而是真正与珠三角及广东地区（甚至扩展到全国）整个经济发展、产业结构相结合，以软件科学服务信息化，以信息化服务制造业和其他支柱产业。

在产品质量认证领域，深圳天祥（Intertek）质量技术服务有限公司和莱茵技术监督服务（广东）有限公司较具代表性。二者均源自欧洲，一自英国，一自德国。天祥为广东的玩具制造出口行业提供测试、质检、工厂评估及企业社会责任监督等一站式服务，而莱茵则以其超过130年的经验，帮助青岛海尔、科龙、格力电器、美的、TCL国际电工、格兰仕、广西玉柴、三一重工、三联重科、长沙远大、四川长安、宗申、力帆等大型企业提高再出口产品安全国际认证、质量管理体系认证、汽机车产品安全认证与轨道技术安全评估、管理技能等方面的水平与能力。

这两家从事产品质量认证的企业，在软件科学园的各种文件中多次出现，足证园区扶持贴近产业结构特色企业的方针。如果说天祥更为与地方产业结构（制造业）紧密结合的话，莱茵则已立足广东、服务范围拓展到全国。

人才培养着眼未来

出自广东的《喜羊羊与灰太狼》风靡中国，给发展动漫的产业战略打了一剂强心针。2008年7月，广州市发改委和软件办联合下文，认定广州开发区软件和动漫人才培养培训基地为政府资助的软件和动漫人才培训中心。基地以广州火炬中心牵头建设，由广东软件科学园建立专业的软件培训平台承担软件人才的培养培训任务。

早于动漫人才培养基地一年成立的IBM软件培训中心，是IBM在中国成立的第二个软件授权培训中心。成立至今，该中心在广东外语外贸大学、广东工业大学、广州大学等多间院校举行大型的技术讲座和交流互动，并与番禺职业技术学院合作建立软件测试和质量管理技术培训中心，共组织了IBM软件工程实训项目、软件人才外包培训、大型软件外包讲座等各类形式的培训班40场次，参加培训人员超过2400人。

在整个人才培养体系中，最面向未来的是针对高校毕业生的实训计划。2008年2月，广东软件科学园、中国软件产品出口协会和Digital River合作设立《大学生软件出口人才实训工程》人才培育项目，该项目以面向欧美市场的共享软件的设计、评测、销售为实践内容，与高等学校计算机软件、电子商务、美术、外语等学科的课程体系和课程内容改革相结合，展开对大学生以适应软件出口外包为目的的人才培训。这也呼应了发展软件出口的政策导向。

重视小型企业孵化

目前，由广东软件科学园、广州软件园和南沙资讯科技园共同组成的“广州软件园”，已经成为国家级软件产业基地之一。在谈到园区未来规划时，文俊伟说，园区将更重视对小型专业企业的孵化工作。广东软件科学园还挂着“国家863软件专业孵化器广东基地”的牌子。基地与科学园于2004年同时正式挂牌，是国家科技部与广东省人民政府的共建项目，旨在为国家863软件成果、软件企业提供优良的孵化环境及技术支撑环境，并突出通信、信息家电产业服务特色。

七幢大楼和12000平方米的绿化面积，已占满首期100亩用地。文俊伟和他的经营班子，正在想办法拓展园区范围和对外合作。“下一阶段的工作中，软件园将以更加解放的思想，更加开放的姿态，更加创新的机制和体制，积极主动地实践广东自主创新的经济发展要求，用战略性的眼光积聚更多高端资源；继续拓宽服务领域，加大公共服务力度；充分发挥软件外包产业服务平台优势，推进粤港软件外包深度合作；关注信息安全核心技术，推动信息安全产业发展；为区域中小企业提供更加优质、全面的服务，切实为广东软件产业发展作出贡献。”

广东软件科学园坚持走科技与经济相结合、经济效益与社会效益协调发展的原则，一手抓园区“硬环境”建设，一手抓“软环境”建设。

一个老拥趸眼中的极限编程之父

——记2009年Kent Beck中国行

■ 文 / 莫映

小引

还记得2002年，自己第一次从杂志上了解到敏捷方法这样一个新鲜事物，当时着实心潮澎湃了一番，寥寥几句敏捷宣言，仿佛道出了憋在心里许久未曾吐露的心声；同样是那一年，国内的人民邮电出版社出版了一套极限编程系列丛书，这标志着作为敏捷方法诸多流派中极具影响力的一支——极限编程（eXtreme Programming），开始逐渐步入了国内众多普通软件从业者的视野。当然这其中也包括了我，那是我第一次了解到这样一种与以往所认识的传统软件工程方法迥然不同的新方法学。在这一系列丛书中，包括了被人们称为XP纲领性宣言的《解析极限编程——拥抱变化》一书，该书的作者是极限编程的缔造者Kent Beck先生，从那时起，我便加入了XP拥趸的行列，与国内软件同行们一道开始了自己的实践之旅。也是从那时起，Kent Beck这位极限编程之父，成为了我心目中的偶像。

光阴荏苒，转眼7年过去了，虽然这些年里成就不多，但大抵我还可以算作是一个名副其实的XP老拥趸。但凡拥趸之于偶像，心里不免都会有种隐隐的期待，希望有朝一日能够近距离的看到这位传奇大师。出乎意料的是，这样一个不经意的愿望竟会在7年后成真——就在上个月，我听说Kent Beck要来中国讲学，作为他的首次中国之行，他将为国内软件开发者奉上为期一天的讲座，讲述近年来的研究心得。千载难逢的机会岂容错过，于是我便迫不及待地报了名。

响应式设计（Responsive Design）

响应式设计（在尚无确切译法之前，姑且这么翻译）——是Kent Beck本次讲座的主题，也是他近几年来主要的研究项目。记得此前在AgileChina的邮件列表里，作为活动主办方负责人之一的霍泰稳，曾经这样描述Kent Beck的这次讲座：



本文作者与Kent Beck（右）

“在我和Kent Beck的电话沟通中，他少有激动地告诉我，最近终于搞明白设计是怎么一回事了，有点醍醐灌顶的感觉。这也是其35年来的精华总结，据他自己说，如果将这次的讲座做个比喻的话，你就想想十几年前听我讲极限编程的那一小撮人现在的感受吧。”

在本次讲座中，Kent Beck首先介绍了响应式设计这一研究项目的背景，随后便开宗明义地指出，在他看来，设计——或者说，响应式设计——的目标，是为了实现持续稳定的交付软件功能（Steady flow of features）。在这样的前提下，过早或过迟地作出设计决策，都将会阻碍这一目标的达成。一个大家广为熟知的例子便是所谓的超前设计（Up front design）：一个预先设计好的架构很可能会在系统的后续开发阶段中被证明很难适应需求的变更。对此，Kent有一个形象的比喻：既有的软件架构就像是一个预留了一系列方形和圆形槽口的模子，而当我们尝试加入新功能时，却发现需要植入的是一块三角形积木，这并不符合任何一个现有的槽口，于是我们就需要停下脚步来修改模子，稳定交付软件功能的目标便无法达成。

事实上，要达成持续稳定交付软件功能的目标是充满挑战的，我们不仅要考虑如何设计的问题，更重要的是要考虑何时进行恰当的设计，以及如何在众多约束中做出权衡。从响应式设计的观点来看，设计是软件开发过程中的一种持续性投入，而非一次性的初识投入。

为了帮助设计者进行设计决策，响应式设计提出了价值观（Values）的概念，亦即设计者在设计时应该着重考虑的一些因素，例如：注重反馈（Feedback），深知软件是为人（Humanity）服务的，了解并正确对待软件设计中出现模棱两可现象

（Ambiguity）所带来的负面影响，等等。这些都将影响设计者的设计动机，从而影响设计方案的选择。

只了解价值观还不够，因为它恐怕太过含糊和抽象了，所以我们还需要一些原则（Principles）对我们进行引导。在响应式设计中，安全前行（Safe Steps）便是很重要的一条原则：平衡软件开发中的效率与风险，选择以一种小幅而安全的步序逐步交付软件功能。只有这样才能实现“Steady flow of features”的目标。

不仅如此，作为本次讲座最为核心的内容，Kent在他过去对设计实践的研究中还发现，所有的设计行为，无论具体业务领域是什么，最终几乎都可以被归纳为少数几类设计策略：

1. 跳跃策略（Leap），如果一切了然无疑，那就可以即刻采取行动，付诸实现；

2. 并行策略（Parallel），如果对设计没有疑议，但却无法确信是否能够做到“安全前行”，为了保证系统能够照常运转，那就不妨在保留原有设计的同时引入新的设计，然后逐渐过渡，直至我们可以放心地去除原有设计而不必担心安全隐患。这种策略时常会在遗留系统的维护期间被采用；

3. 垫脚石策略（Stepping Stone），如果我们对作出何种设计决策并不清楚，那就不妨快速建立一个原型来帮助我们实现原本困难的设计。垫脚石可以表现为各种形式，例如：某个公共组件、服务、框架等。或者，更为经典的一个例子则当属领域特定语言（Domain Specific Language，DSL）了；

4. 简化策略（Simplification），如果问题过于复杂，以至于我们无法看清最终的目标，或者要达成这样的目标代价高昂时，那就不妨简化问题、减少需求，直至我们可以“安全前行”为止。

最后，假如你果真无法找到一个妥善的办法，能够顺利地将新功能加入到现有设计中时（读者不妨想想前文所提到的那个三角形积木的例子），那就不妨照加不误，直到日后能积累到足够多的反馈信息，让自己有能力做出正确的判断为止。

讲座的内容还不仅限于此，由于篇幅所限，此处无法一一道来，感兴趣的读者不妨阅读Kent Beck的个人博客，那里有不少他本人对响应式设计的最新观点。

Responsive Design和XP的关系

这一点也是我在听课期间始终感到好奇的地方。如前所述，Kent在阐述他对Responsive Design的理解时，引用了Value和Principle的概念，这些于我而言都似曾相识。因为，在《解析极限编程》一书的第二版中，Kent就是利用同样的方法来阐述整个XP的理论体系的，即：Value作为XP的核心价值取向，支撑着所有的Practice；而Principle则引导人们使用正确的Practice实现价值，避免因误用而走上错误的道路。由此似乎可以看出，XP和Responsive Design实际上有着千丝万缕的联系，至少在阐述方式上，二者是一脉相承的。带着这一问题，我向Kent进行了求证。

关于这个问题，综合Kent Beck的回答以及我的理解：我们可以将Responsive Design视作是XP在某个局部上的深入探究，这一局部便是——设计。以往，XP常为人们所诟病之处，恐怕就在于它的“轻设计”：寄希望于从一个差不多够用的初始设计开始，经过不断重构来修正架构的方法，似乎总会让人为其捏一把汗。而近年来，Kent以他自己的实际行动，在这方面进行了深入的研究。所以有理由相信，Responsive Design应该可以为XP在阐述设计方面所遗留的“缺憾”做一些补充。

关于XP的过去和未来

场间休息时，我得隙拿着一本《解析极限编程》（中文第二版）找到Kent先生签名，之所以选择这本书是因为我与该书的一点特殊渊源。自《解析极限编程》第一版问世之后，Kent Beck并没有驻足不前，而是根据他人的反馈，不断修正自己对软件过程方法的看法，并将这些新观点更新补充到XP中。于是在2004年，该书第二版向读者见面了，而相应的中文版则在一年之后由国内的电子工业出版社引进并翻译。那时我刚好得幸参与了该书中文手稿的校审工作，也算是国内较早拜读该书的少数几人之一了。相比于第一版，这一版在内容上有了很大的变化，更为系统的思考与详实的论据，让我深信XP作为一种理论方法已经日臻成熟，也愈加为Kent深厚的功力所折服。我依然记得初读手稿时激动的心情，反复阅读三遍过后，仍是爱不释手。而现在我手中所拿的这本书，便是事后出版社寄来的样书。

当我告诉Kent此书的来历时，他笑着跟我说，他非常喜欢这本书的第二版，因为这一版在思想上更加成熟，言语间用签字笔轻轻在扉页的书名上画了一道印，随后便欣然签上了他的名字。后来我问他，何时打算出第三版时，他斩钉截铁地回答说，现在尚且没有这个打算，他补充解释道，因为方法学也是一门实验科学，需要依据大量的试验进行研究，而近期他公开的主要研究方向依然是Responsive Design，所以，短期内在这方面不会有大的更新。

小结

为了尽可能准确地把握响应式设计的要义，撰写本文期间，笔者也查阅了一些网络资源，让我颇感兴趣的是，通过翻看Kent以往的文章和最新的博客，我们可以看到这项研究工作并没有就此止步，正如他数年前研究极限编程一样，这位孜孜不倦并且乐此不疲的软件泰斗，依然在不断地推陈出新，着实让人心生佩服。最后，让我们共同期待来自Kent Beck先生的下一个新惊喜吧！

风云再起之 CRM

——记微软CRM模板设计大赛

■ 文 / 丁莹

2009年9月，苏州风云网络联合微软中国，举办了“设计——我能，微软CRM模板设计大赛”。金秋十月，大赛正在如火如荼地进行中。CRM模板设计大赛是基于风云在线微软CRM（客户关系管理）应用集市（Marketplace即沙箱）为基础架构，鼓励参赛者开发出可满足企业或个人日常业务需求的SaaS模式软件产品。

风云在线是苏州工业园区政府与微软共同打造的微软亚太区SaaS孵化平台，同时也是微软大中华地区唯一的微软CRM托管运营平台。据风云在线总裁戴晔先生介绍说，此前苏州工业园区政府正在积极行动，希望为中小型成长企业提供高端专业化的服务，这个想法与微软CRM的发展战略在很多方面不谋而合，也因此很快促成了风云在线的诞生。

与微软传统CRM业务不同的是，风云在线的平台更侧重于向中小型成长企业提供在线的解决方案，解决中小企业信息化过程中因成本过高而产生的障碍。通过风云在线对硬件、软件、开发方面的投入，企业可通过按月租赁的方式，很快上马CRM系统。通过这种SaaS模式的CRM，企业可大大降低投入成本，迅速迈上信息化的快车道。

让更多企业走便捷化CRM之路

据微软中国CRM托管业务负责人、CRM解决方案经理徐曦先生介绍：截至2009年7月，微软在全球的CRM企业客户已达2万家，用户数更是达到了100万以上。从2006年5月微软发布CRM中文版后，中国客户数也是连续三年三位数增长达到数百家。这反映出国内企业对CRM的认识在逐渐提高到一个更新的层次。

“适合我的员工，适合我的业务，适合我的IT的环境”作为微软Dynamic CRM的三个核心价值体现了微软Dynamic CRM的发展是朝着易用性、适合企业的业务、满足IT环境的集成、扩展和维护三个方面前进。在合作管理模式上，微软采用了基于合作伙伴的业务模式，并且致力于向企业客户提供完善可行的解决方案，由此来帮助企业成功发展。



微软中国CRM托管业务负责人、CRM解决方案经理徐曦

微软最新发布的Dynamic CRM 4.0不但从功能上有了很大提升，而且突出体现微软S+S战略，融合了SaaS的运营模式。以SaaS模式运营的CRM其最大的优势体现在方便快捷方面。首先，其使用简单，可以随时注册、随时开通，这样就绕开了繁琐的部署阶段；其次，对客户来说投入变低了，传统的CRM可能需要开发或者购买一个CRM软件，这可能需要一次性投入十万甚至上百万；而在SaaS平台上，一个用户一个月可能就是两三百块钱；第三，企业用户通过OutLook或者Web的方式，可以直接访问到CRM系统内的数据；第四，容易定制和扩展，针对企业来说，目前平台上提供了多个模板，针对开发者来说，他们可以直接在平台上通过自定义工具满足市场上各类企业的业务需求。

风云在线这样介绍道：未来国内的CRM系统会朝使用简捷、快速应用以及可灵活定义逻辑这样一个方向去发展。而SaaS的出现一方面让企业降低使用成本，另一方面使企业的使用范围更广、更灵活。

做更完善的CRM解决方案

为了让市场了解微软Dynamic CRM更多的开放性、构建不同的解决方案，并通过这样一个活动来促进微软CRM开发的技术和相关的解决方案的市场的认同度，微软携手风云在线举办了此次比赛。而风云在线也想通过这样一个活动来引入与之相关的解决方案：一方面可以汇集全国致力于应用开发的技术人员，通过微软的CRM平台展示自己的行业经验、技术



风云在线总裁戴晔

实力和设计理念；另一方面最终用户通过对产品的应用，可以提出行业性的建议，有利于平台上丰富各种行业的应用，可以扩充我们的行业解决方案，形成一个设计开发者与最终使用者的交互的平台。

戴晔说："我们现在的想法是：希望通过这次大赛，把我们的应用模板丰富化、多样化，所以我们需要更多的开发人员参与进来，充分展示出自己的行业开发经验。因为开发人员遍布全国各地的各个行业，我们需要把这些行业经验整合起来，融入到整个软件管理系统当中去，用管理软件的思想去体现这些行业经验。同时从用户的角度去考虑，如何通过简单的软件应用来帮助用户解决日常管理中的问题。因此这些与赛的开发者，不能仅仅从纯技术的角度去设计，因为从纯技术的角度去开发，会导致一些局限性，或者是很难把自己的行业经验融入到其中。我们希望入围的产品，能真正给企业实际采用，并带来效益，从而达到多赢的一种态势。"

吸引眼球的创业基金——三年一百万

对于参赛作品的评判标准，风云在线与微软达成了一个共识：第一，在市场上有广泛的接受度；第二，能够很好地体现CRM的业务理念和核心价值；第三，对微软的核心技术是否有很好的应用。而通过了这三个评判标准的获奖作品，风云在线将给予其作者一定的奖励。针对这次的比赛的优胜者，风云在线将提供三年一百万的创业基金。另外，微软方面针对于不同级别的获奖作品，除了提供相应的奖品奖金之外，还将提供获奖者参加微软Dynamic CRM认证考试费用、申请成为微软CRM MVP、参观微软技术中心、推荐到微软CRM相关实施合作伙伴等的不同奖励。因此，这次比赛吸引了众多.NET高手、微软CRM技术顾问和业务顾问和ISV（独立解决方案供应商）踊跃参加，由此达到了一个知识共享和迁移解决方案的目的。

本次大赛除了物质上的奖励外，在实际设计开发中，风云在线也为广大的开发者提供了很多便捷的帮助和支持。首先，风云在线专门提供了一个微软Dynamic CRM模板设计教程，可以系统全面地帮助初学者了解和掌握CRM的相关知识；在参赛的流程中也提供线上开发和本地开发两种模式，参赛者可以很方便地进行选择；考虑到部分参赛者还有个学习期，所以主办方将作品提交的期限拉得比较长，给与赛者留出比较充分的设计时间；风云在线还提供了400客服电话以及技术专线，接受参赛者的咨询。尤其要指出的是，对于行业经验丰富，但是开发经验有限或开发经验丰富，行业经验有限的参赛者，风云在线将会把他们组合在一起，把双方的优势都发挥出来，这样一来就可以设计出更好的作品。

对于所有提交作品的参赛者，还有机会参观微软技术中心；此外提交作品的幸运参赛者，还可能得到微软提供的微软工程师的考试券。一旦作品入选，还可以直接在平台上向企业用户提供应用，并进行销售分成，所以针对模板开发企业或者个人，这可能是一次非常好的创业良机。此外，未经参赛者同意，

微软CRM模板设计大赛调查数据

数据来源：CSDN网站（2009年10月）

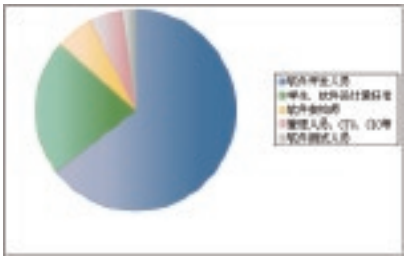


图1 关注大赛人群分布图

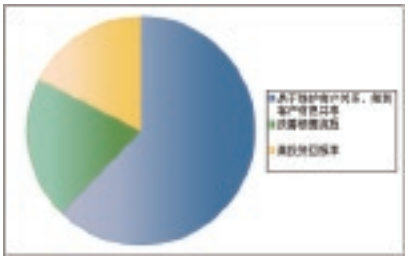


图2 实施CRM能够为企业带来的优势

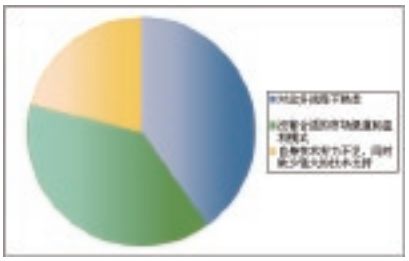


图3 国内企业（个人开发者）在进行CRM产品开发时可能会遇到的问题

风云在线不会以任何形式使用参赛者作品，风云在线愿意和优秀的开发者合作，共同向企业推广其参赛作品。并将开发人员的行业经验，形成一套解决方案，如果参赛者愿意合作，经过授权后，风云在线会把他的作品放在平台上向外进行销售，同时进行销售的收入分成。

虽然CRM系统现在仍然让许多企业客户"爱恨交加"，一方面对于企业中客户及销售流程等方面的管理，这是一项必需的投资；但另一方面，对CRM系统的不够平民化又感到无可奈何。但随着这次微软CRM模板设计比赛的进行，相信CRM将变得更加平易近人，为用户提供更多部署方式的选择，并通过对业务流程的全面管理给企业带来更多的收益。让我们在这次的CRM设计大赛中一同感受CRM新一轮的崛起！

掀起 OPhone™ 平台开发的红盖头

■ 文 / 付江

随着八月底中移动 OPhone平台上线，国内移动应用开发者的创作激情被再次点燃。OPhone能给开发者带来多大的展示舞台和前（钱）景？其独到优势何在？如何才能获取更多的技术资源支持和开发经典案例？

为解答诸如此类的疑问，10月15日，由中移动主办、播思通讯协办的“OPhone平台2009金秋之旅”面向开发者的第一场培训在北京诺福特三元酒店拉开了帷幕。即便记者早已深知当今移动开发的火爆，但当天近300名国内一线资深个人开发者和独立软件开发商 (ISV)的表现还是让记者惊讶。面对主办方特邀的演讲者，现场参会者充分表现出了程序员不畏权威的本色，尤其是互动环节所发出的咄咄逼人的提问，以及与演讲者争锋相对的深入探讨，都让人感受到开发者们对移动应用开发的关注。

API开放力度大、功能强大

巡讲开始，来自中移动研究院的项目经理詹建飞带来了《OPhone平台介绍》的演讲，主要谈及了 OPhone平台技术特性、体系架构、开发流程、应用程序模型以及Mobile Market等方面内容，他着重强调了 OPhone平台的开放性。其中一方面表现在开放 API力度大，功能强大。对于过去用 J2ME做移动开发中常常遇到的诸如屏幕尺寸控制、内存扩展、多媒体格式支持、Socket支持等问题，由于 API开放力度不够往往给开发带来很多麻烦。而对于 OPhone平台，底层提供了非常丰富的库，通过JNI层将大部分 API开放出来，因此开发者几乎可以碰触到所有底层接口，并且可以在开发过程中查看到OPhone的源代码。其次，OPhone开放的 API对待所有的应用程序都是平等的，只要内置的应用程序能使用的 API，第三方开发商就同样能够使用。


座无虚席的会场

OPhone开发者社区SDN

接着，来自于播思通讯公司的软件工程师李若飞给大家带来了OPhone平台开发 SDK的介绍，并着重谈到了 OPhone开发者社区 SDN(www.ophonesdn.com/)目前的运营情况。李若飞重点介绍了 OPhone SDN。开发者不仅可以在这里下载到最新的 SDK，查阅到OPhone官方的开发文档，阅读开发实例，建立自己的软件项目，还可以通过论坛与其他移动应用开发者交流开发经验。而论坛中的版主大多也由中移动研究院或播思通讯的资深技术经理担任，他们会积极上论坛参与讨论和回答开发者提出的问题。针对与会者提出的利用OPhone平台开发诸如联网的在线阅读软件，在模拟器上运行正常，放到真机上就无法联网的问题，李若飞透露播思通讯的工程师们已完成关于联网的 DataConnection方法的文档，将很快放到OPhoneSDN上供开发者阅读下载。

最后来自于中移动研究院的项目经理杨晓华为与会者带来了关于 BAE/JIL Mobile Widget的介绍。Mobile Widget作为在移动终端运行的实现某些特定功能的小应用程序，相对于传统的用 J2ME或 C++开发移动应用，由于其基于标准 Web语言从而大大降低了开发门槛。而且可以在任何安装了Mobile Widget引擎的移动终端运行，真正实现了一次开发、跨平台运行的特点，而不再拘泥于移动终端类型。目前已经实现支持 Limo、OPhone、Symbian、Windows Mobile多个智能终端平台。此外它还改变了移动终端过去的互联网访问模式，可以直接一键式快速访问本地和网络资源。Mobile Widget包括的实际应用有查看股票信息、每日天气更新、日历提醒等。据悉，本次“OPhone平台 2009金秋之旅”将历时2个月，横跨9座城市、遍布8所高校，与高校学生、一线开发者共同分享OPhone平台的精髓。



幽　　默

开发部轶事

开始，上面来个人叫你开发一辆自行车。他会说：就一辆自行车，你看多简单呀！就两轮子，加一个三角架。你问：有什么要求？他通常回答：没什么要求，就是一辆自行车嘛！最后不忘问你一句：明天能好不？

第二天，也可能过几天，他会来问你：开发得如何了？好了吧？你会说：还有点，过几天就好。这时他就会在你身边看，不时提点意见，最后说：干脆，你给它装个发动机，做成摩托车吧！你说：啊！这个难度高了吧？还要找变速箱、离合器的资料。他会说：不怕！要的不是很急，加个发动机而已，又不是很难。然后你就开始去找资料。

再过几天，他又来了。问：快好了吧？没等你回答他又说：我回去想了一下，反正都加了发动机、变速箱、离合器，干脆再多加两个轮子，开发成一辆汽车好了。然后走开，留下你愕然而立。

再过段时间，他再来，说道：这几天我与他们讨论了一下，觉得把汽车上多加一对翅膀，做成架飞机，这样爽点，怎么样？这时你已经没有多少言语了，不管他，做自己的事。

再过段时间，终于他又来了，同行的还有老板和几个客人。他们一起来参观你的作品。老板对客人大肆吹嘘：怎么样？我们的开发能力，世界一流吧！那个谁，在飞机上再多加几个火箭助推器，做成个飞船不是更好？！你愕然：啊！？

这时，先前那个他马上跳出来：老板英明，算无遗策，千秋万代，必一统江湖……然后他们相互拍马，吃饭唱歌泡澡也顺便泡MM去了，没你什么事。该干啥还干啥吧。

再过几天，客户的订单来了：飞船xx艘，10天内交货！


seven
SEVEN
Seven

Windows 7: 再现辉煌

■ 策划 / 本刊编辑部

酝酿已久的Windows 7终于在2009年10月发布了。虽然有Vista的阴影，Windows 7的表现却得到了很大意义上的认可。其RC版本在互联网上广为流传，并经过了大量用户的实际使用。期间甚至有专家认为：“Windows 7完整地满足了它所针对的用户群体需求。”

这是一个很高的赞誉，尤其对现阶段的微软产品而言。就像在社区里讨论已久的那样，微软到底还能与谁竞争？今天的微软，要打败的对手只有自己。如何去寻找看起来已经饱和的桌面客户端用户需求？如何将现有的技术进一步完善和改进？如何通过产品与服务整合的战略占领新时代的制高点？相信这些问题都是微软苦苦思索的。

微软在这三年中并没有止步，他们依然完成了非常惊人的工作。在操作系统端，Windows 7的出现无疑是一个答案——数以万计的驱动程序、数以万计的应用程序兼容性测试、数以千万行计的代码编写……

然而，它是否会为今天的IT产业带来新局面？面对网络操作系统的轮番猛攻，桌面操作系统的存在价值是否能够得以延续？以手机平台为核心的客户端设备是否会对桌面操作系统带来冲击？这些问题的答案都将随着Windows 7的来临逐渐揭开。

也许这些问题都没有标准答案，但在本期杂志中，您将看到来自各个角度对Windows 7乃至整个客户端操作系统平台方面的分析与评论。在我们邀请的专家中，有来自微软、英特尔、英巴卡迪诺、雅虎中国、支付宝以及金山软件的资深开发者，相信用他们的独到观点，定能让我们用不同的眼光看Windows，看微软。

第七代 Windows

—— 再现微软时刻

■ 记者 / 欧阳璟

北京时间2009年10月23日凌晨（美国当地时间10月22日上午），微软正式向全球发布其新一代的客户端操作系统产品Windows 7。鉴于其上一代产品的失败，这一版Windows的发布，可以认为是微软洗刷过去三年来耻辱的一战。尽管中国的操作系统市场由于盗版问题，一直让微软特别头疼，但在这个特别并且重要的日子，微软仍然派出Windows产品部副总裁Mike Nash来到中国，为我们揭开Windows 7的神秘面纱。在这一刻到来之前，我们不妨先回顾一下过往的微软Windows时刻。

微软的时刻

天之骄子——Windows 95

1995年8月24日，开发代号为Chicago的Windows第四个版本正式面世。时任微软CEO的比尔·盖茨似乎知道这个产品即将取得巨大的成功，因此他毫不吝惜任何形式的宣传，耗资1亿美元，在帝国大厦的顶层竖起了一个日后几乎每个PC用户都非常熟悉的霓虹灯广告。从此，微软帝国正式建立起来。

这个 16/32位混合操作系统，不仅仅创造了软件销售的历史，更对整整一代人产生了深远的影响。用鼠标点击的计算使用方式，成为人们日常生活和工作的标准动作；记住你的文件放在 C盘还是 D盘，成了很多人使用计算机的基本模式；每当坐在电脑面前开始工作，我们就试图去寻找左下角的开始按钮……

不论从哪个角度看，Windows 95都给整个计算机乃至IT产业作出了巨大的贡献。微软因此得到了回报，那就是在之后的10多年时间里，始终引领着PC客户端技术的发展。“Windows 95的图形化界面，让很多人可以开始使用计算机，这是它最大的贡献。”Mike Nash在评价这款产品的时候提到。

盛世升平——Windows XP

6年之后的同一天，也就是2001年8月24日，开发代号为“Whistler”的 Windows XP面市。也许当时谁也没有想到，这款产品将在未来的 8年时间里始终成为微软最重要的产品。直至2009年8月底，Windows XP一直是世界上最多人使用的PC操作系统，这款产品拥有高达71.79%的市场占有率；2006年12月，Windows XP有最高峰的85%市场占有率（数据来源：维基百科）。

由于采用了更加稳定的 Windows NT 5.x内核，XP提供了更加友好的用户界面，并拥有更出色的性能。同时，Windows XP内置超过 10000多个设备驱动程序，几乎兼容了整个PC业界所有的硬件产品。这款微软历史上最成功的产品，也为微软稳固了软件帝国的地位。

再续辉煌——Windows 7

转眼之间，已经到了2009年10月22日，Windows 7还没有正式上市，其 RC版本已经被不同的媒体和分析机构研究了个底朝天。与以往充满批评的声音不同，关于Windows 7的市场反响非常正面，大多数使用过 Windows 7的人都对这款新时代的客户端操作系统很满意。然而，Windows 7能否继续保持辉煌，并让这个帝国继续强大下去，

功能特点	MS-DOS	Windows 3.1	Windows 95	Windows 98 \ Windows Millennium Edition
GUI界面	X	√	√	√
Luna界面	X	X	X	X
Windows Aero	X	X	X	X
支持MS-DOS软件	√	√	√	√
Windows Flip 3D	X	X	X	X
内置驱动程序数	?	?	?	?
系统内核	MS-DOS	MS-DOS	MS-DOS	MS-DOS

还需要时间和市场的证明，但是至少她已经创造了一个良好的开端。

革命还是修正？

被问到最多的一个问题依然是：Windows 7是不是微软历史上具有革命性的产品。答案很多，但是至少在很多专家的眼里，这并不是一场革命。而且，不少专家甚至在Windows的发展历史中发现了一个怪现象。

Windows 95创造了奇迹，但从产品角度而言，其升级版Windows 98却比其前任要好得多。同样，Windows 2000作为Windows NT内核技术正式成为“东宫”的第一代产品，相比其升级版的Windows XP却远远没有后者成功。

如果我们深入了解 Windows 7就会发现，这个沿用了Windows NT 6内核的操作系统，看起来更像是一个Windows Vista的升级版。Mike Nash对这个问题并不避讳，但他同时认为这款产品有其革命性的一面：“从产品本身看，新产品既是对原有产品的发展，也从一定意义上创造了革命。比如在安全性和用户界面上，Windows 7发展了 Windows Vista，而在操作和控制上，多点触摸技术的出现就是一场革命。它带动的不仅仅是人们使用操作系统方式的革命，而且带动了整个IT领域软硬件的产业变革，这其中还包括大量微软的合作伙伴和ISV。”

尽管多点触摸技术的发明者并不是微软，但是作为最有影响力的PC软件提供商，微软是有实力将这种新的模式推广到全世界的，这就像当年Windows 95并没有发明鼠标一样。

当然，不是革命的理由在微软看来也非常充分。多数时候，革命代表了牺牲，而在PC客户端操作系统领域再无竞争对手的微软，要革自己的命显然不现实。“相比过去的Windows 95而言，今天的时代有很大不同。过去我们并没有这样一个庞大的用户群体，而今天，Windows 7要面对的是数以十亿计的用户群体，要满足这样一个巨大的用户群，做任何一点改变都是非常困难的。”Mike Nash解释道。

换句话说，原有的Windows需要由新的Windows来打败，这不仅需要创新、技术，也需要勇气。Mike Nash说：“幸运的是，我们几乎做到了这点。不少使用Windows 7 RC版本的用户甚至感觉这套系统是为自己单独定制的。”



微软公司Windows产品部副总裁Mike Nash说：“Windows是技术力量和商业力量的结合。”

操作系统新战略

不管怎么说，比尔·盖茨当年的愿景已经实现。今天，大多数家庭和办公场所都有正在运行着 Windows 操作系统的个人电脑，这是微软最初定义的战略上的成功。然而在互联网时代，人们的 PC上是否还需要 Windows？不少开源精神的拥护者和互联网的先行者们高呼不再需要 Windows的同时，我们依然应该看到Windows创造的历史。

Mike Nash说：“Windows的成功，不仅仅代表了技术的力量，也代表了经济增长的力量。Windows 7的出现，是微软长期投资的直接表现。作为在客户端操作系统领域的微软，会长期持续投资于桌面客户端领域。这不仅仅包括 Windows，也包括Office、Bing以及各种各样微软的服务器产品。”

尽管今天互联网正在以爆炸式发展，尤其是伴随着最新的RIA技术，它正在以空前的速度增长着，但是向后看10年时间，操作系统仍然有很大的成长空间。淘宝 UED部门的资深用户玉伯在和记者的沟通中提到：“RIA跟操作系统并没有冲突，他们提供不同的功能。Web的兴起并不是对Windows的挑战，因为操作系统有操作系统要做的事。通过试用Windows 7，我看到它非常漂亮地完成了其历史任务。”

其实，操作系统的作用就是为了让用户更方便的使用，如何想办法让Windows用户尽快迁移到新平台，并让他们真正了解到新平台对用户体验的提升，才是Windows应该完成的任务。

	WindowsNT 3 \ WindowsNT 4	Windows 2000	Windows XP	Windows 2003	Windows Vista	Windows 7
	√	√	√	√	√	√
	X	X	√	√	X	√
	X	X	X	X	√	√
	X	X	X	X	X	X
	X	X	X	X	√	√
	?	?	10,000±	?	28,000±	?
	NT3.1、3.5、3.51、4.0	NT 5.0	NT 5.1	NT5.2	NT 6.0	NT 6.1

*图表来源：维基百科

Windows 7 中的架构视点

尽管Windows 7涉及的新特性不少，而且对既有功能有一定增强，但从技术表象看Windows Vista似乎更像一个Windows 7的beta版。一般提到Windows 7，大家普遍关注它的界面中可获得性(accessibility）因素、相对 Vista的性能改进、联邦搜索能力、UAC，此外能够同时兼容多版本操作系统的能力也被称道。

■ 文 / 王翔

对于架构师而言，性能的改进、广泛的兼容性、跨存储资源的搜索、可获得性提高等都不是简单的技术改进，如果后台的技术架构不做出调整，对于 Windows 7这样一个复杂的系统应该难于达成，毕竟任何最终用户能够感触到的内容都是经过数层组件调用和封装实现的，由于技术历史的沿革其中很多调整相信会涉及整个产品架构组织。

本着 By Architecture〉By Design〉By Patch的原则，这里我们用架构师常用的技术思维考量一下Windows 7，分析它能为我们未来基于Windows 7平台设计新的应用架构提供哪些支持。考量内容包括如下方面。

架构思路和产品技术协议

您是否习惯于分层架构设计？不过这次Windows 7却把我们的目光吸引到一个叫Min Win的最小内核上。

以往我们总是抱怨Windows太大，捆绑的东西太多，很多操作系统狂热者认为Windows不如某些操作系统那么纯粹，甚至Windows也在欧洲也多次被强制执行。但真把Windows 7放在一个25M的空间上，启动时可以看到图1中一个用字符组成的Windows logo时，您会发现原来不是Windows庞大，而是我们自己对Windows的深入了解不够。

这个Min Win并不是简单的本地操作系统环境，他提供了TCP/IP协议的支持，能够提供简单的设备驱动，还能提供后台任务和服务的支持。Min Win提供了一个基于微内核架构的环境，抹去了分层架构上中层的所有扩展机制和服务，围绕它您甚至可以重新构造一个独立的后台操作系统环境。相信图2您已经比较熟悉，但Min Win其实只包括其中 Kernel Mode中的一部分加上非常基本的用户态支持。Min Win给架构师什么启发和参考呢？



图1 Min Win的ASCII 启动画面

- 如果您的系统过于复杂，现有的上层实现必须有更好的、更彻底的底层技术支持；可以考虑专注于孵化一个新“内核”的设计；
- 如果您从事了一系列项目后发现技术设计部分无非“那点事”，这时候可以把这些抽象的“那点事”做成一个核心；
- 最后，为什么不做成分层而是微内核呢？因为“核”内包括的功能、特性足够稳定，而且组件间存在普遍的交互关联（inter - related），经过多年积累您发现连接他们最有效的性能组合，这时候就可以把它们编译成一个“压缩饼干”。

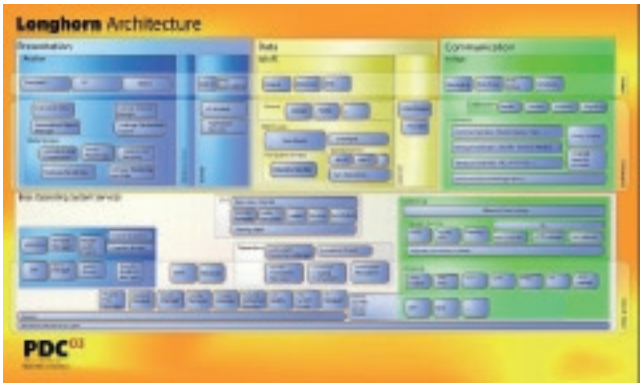


图2 Longhorn的技术架构

Windows 7中体现出微软对技术标准、开发体系的态度相对以前有进一步的转变，不仅处处向 PHP社区伸出橄榄枝，对自己既有平台的开发者也提供延续性的支持。相对 PHP，之前风传 Visual Basic 6要被 Windows 7除名的消息更具震撼性，但微软发布的更新后的支持协议表明 Visual Basic 6将获得Windows 7生命周期内5年主流支持和5年扩展支持。

对象化分析和设计

以往开发Windows平台产品与开发业务应用产品的不同主要体现在调用对象上，前者采用的大部分是用 C/C++调用的各类结构体、联合体、API方法，后者则更多采用对象化系统。随着.NET技术的深入，Windows Platform SDK中越来越多的内容已经封装为 .NET类型，这样开发和设计人员无须面对复杂的API重复封装并可能不够高效的实现一些Windows平台控制对象。

以Windows 7中的 multi touch为例，如果封装

API的方式实现，在不考虑 COM的“八股”内容外，您大概需要数百行代码封装一个多点手指触动屏幕后抛出回调句柄的基础类型，但在 Windows 7平台上，相关类型已经被封装到 Windows7.Multitouch及其子命名空间中。

那么对于架构师从对象化设计角度看，Windows 7提供的这些有什么益处呢？（见图3）

关注点分离（SOC：separation of concerns）、设计领域模型分离。

其中，不难发现封装的 Windows7.Multitouch同时肩负下列角色：

- 一个 Front Controller，所有对 Multi touch的调用都集中到封装的类型内部，以后即便需要扩展其他响应内容，也只需借助对象化的委托和事件获得通知，无须新的逻辑显式调用API。
- 一个View Helper，对于客户程序而言，所有Windows 7 multi touch的内部实现通过对象系统屏蔽起来，他负责把客户程序调用分配到不同界面元素。
- …

上面只是 Windows 7对象体系的一个缩影，基于新的 Windows平台您可以将诸如安全性、客户端访问控制和资源约束、界面、地理信息位置感性等 NFR（非功能性需求）的“子系统”分离出您的应用，业务系统与这些平台子系统仅通过 Windows 7的 managed 类型接口进行交互，如果您需要扩展 Windows 7的相关内容也可以借助 IoC把新的逻辑反转到平台。

业内也许有很多的 IoC & DI框架，但您是否考虑过把 Windows本身当成个最大的IoC & DI框架来用呢？

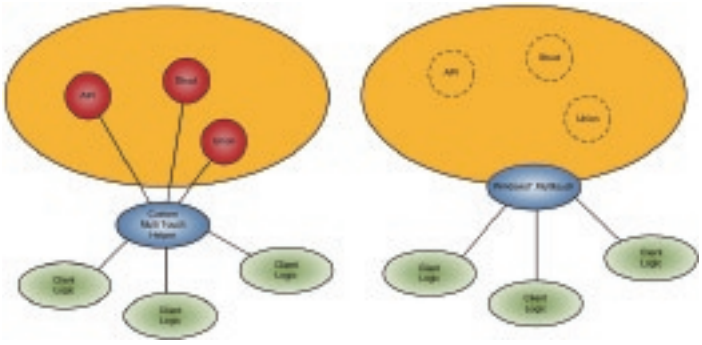


图3 增加Windows7.Multitouch前后的概念结构

.NET技术应用

作为团队的技术导师，架构师需要熟悉开发平台，对于Windows应用的架构师而言更是如此。Windows 7开发平台的优势在于其全面、复杂性在于其庞大、难度在于对其技术的选择上。时至今日，找到合适的 .NET技术和开发框架（官方的、开源的、组织积累的）完成业务功能也成了一个挑战，毕竟完成一件工作我们需要集成内容很多。

不考虑待发布的 .NET Framework 4.0平台，我们不妨看一下 Windows API Code Pack for Micro-soft .NET Framework，通过它了解 Windows 7平台增加的.NET功能及其布局：

- Microsoft.WindowsAPICodePack.Controls及其子命名空间，主要包括界面元素控制，包括 Taskbar Jump Lists, Icon Overlay, Progress Bar, Tabbed Thumbnails, and Thumbnail Toolbars、Dialog等；
- Microsoft.WindowsAPICodePack.Net提供对 Windows 7既有及增强的网络互联能力的支持，在提供必要安全保护的前提下，便于开发更适合普通业务用户和商务工作组的应用；
- Microsoft.WindowsAPICodePack.Sensors，这个也许对地理信息和感知设备开发的团队更有意义，借助它可以在3D和2D环境下提供创新应用；
- Microsoft.WindowsAPICodePack.Shell及其子命名空间则提供对纯粹shell交互环境的支持，不仅能够完成简单的命令交互，同时对 drag & drop提供支持；
- Microsoft.WindowsAPICodePack.Taskbar则是对Windows 7中“武装到任务栏”特征的支持，尤其适于一些通知推送、软件配置调整、简单信息监控预警的应用；
- Microsoft.WindowsAPICodePack.ExtendedLinguisticServices服务面向语言分析的应用；
- Microsoft.WindowsAPICodePack.ApplicationServices提供对重启、关机、电池和电源管理等基本计算机控制的管理；

上述功能相信会在 .NET Framework 4.0推出的时候打包发布，而且随着Windows 7后续SP的发布，上层应用开发人员也能持续获得平台特性支持，就如 MSDN参考文献标题中提到的那样——获得“The Best of Windows and the Web”。

设计模式和架构模式

模式对于架构师的重要性不言而喻，毕竟架构师的工作不总是创新，为了平衡资金、技术、市场、时间，巧妙应用已经被实践检验的设计模式和架构模式不失为明智之举。从Windows 7的开发指南中不难发现，该产品设计上处处充满模式的影子，下面我们看3个例子。

首先看一下 Data Redirection，从 Vista 开始 Windows已经对用户访问敏感文件夹、注册表、系统文件有专门的重定向，绝大多数情况下这些对用户和用户应用透明，但同时也引发诸如 UAC变更时无法获得文件的问题。建议的解决方式包括：

- 文件采用AppData和LocalAppData；
- 注册表采用 HKCU\Software 而非HKLM\Software；

其中不难发现“between client and real subject to control access to the real subject and to perform necessary tasks”的影子，所以从I/O层面Windows 7至少在应用着Proxy模式。

其次，看一下 Windows 搜索中增强的 Federated Search，它不仅可以访问 Windows 文件系统，而且也支持对web、专用数据存储的访问，最后所有

信息以统一的界面呈现给用户。

其中不难发现，尽管目标信息 /数据存储接口各异、信息内容格式千差万别，但最终都以简单的列表集中呈现给用户，俨然就是 Adapter模式 + Iterator模式的无编码标准教程。

兼容遗留系统

这里的遗留系统是相对Windows 7而言。

Vista长期无法替代XP的一个主要原因也许在于步子迈得太大，把合作伙伴甩得比较远，进而导致用户无法及时获得相应的软、硬件支持，最后在市场上遇到麻烦。Windows 7显得非常明智，借助虚拟技术实现 Windows xp模拟运行环境。如果不考虑虚拟环境性能开销大、开发团队维护成本大两个风险，看这个决策一举多得：

- 照顾到合作伙伴。
- 照顾到用户。
- 对于微软自己而言，最大的收益就是减少升级Windows 7的市场阻力。

我们不妨把Windows 7的这个策略称为“汽车备胎”。新旧系统的界面是虚拟机，不过区别于传统架构设计的兼容性策略，遗留系统兼容模式位于完全独立的系统，文件/数据、应用和展现完全分离，而我们一般的遗留系统兼容往往需要进行数据同步、逻辑SOA化、Portal整合等。

Windows 7的做法省去用户兼容性开发，但代价是用户的操作体验不够连贯。如果用户“习惯性”切换模式也许就成了Windows 7的悲哀，不过同时也可能催生出一个新的商业市场。

从实现看，Windows 7的这个兼容性设计应该属于过渡阶段。解决类似问题的实现方式可以分为：

- 彻底替换、升级：这个需要看软、硬件提供商的改进情况，用户的 IT预算情况，类似的方式已经在 Vista中碰了不少钉子；
- 采取组件或子系统层面的 Side by side execution，但这将使Windows 7更加庞大，维护成本成倍增加，不合适；
- 采取现有的虚拟机方式，“井水不犯河水”，虽然省事但用户体验不连贯，容易因Windows xp环境的存在，拖累Windows 7好不容易树立起来的优质形象；
- 采取应用虚拟化和自动操作系统适应。微软的 softgrid虽然多次在展会上亮相，但似乎没有集成在Windows 7中，该方式在保证用户体验的同时，可以透明的解决应用程序兼容问题，但代价是微软需要对 Windows 7体系进行大范围修改。

权衡之后，Windows 7的兼容策略也不失为一个多赢的方案，App虚拟化对于Windows 7目前看只 是个美好的远景。

分布式组件和组件容器

随着 Windows平台分布式技术的统一，WCF自然也成为 Windows 7的标准组件运行技术，其中的 IIS 7和 COM+则成为分布式组件容器。

尽管对于大部分应用我们会把 Windows 7视为单纯的 web 客户端或 desktop客户端，但对于开发人员和项目开发阶段而言，依然可以把 Windows 7作为一个本地调试宿主环境，并借助上面提到的多版本虚拟化环境实现跨 Windows 平台的调试。

从分布式组件组成看，Windows 7平台提供了面向 WCF（包括简单 HTTP ASP.NET、WS-*、RESTful、COM+、MSMQ…）的多种分布式组件机制，而且所有信任机制可以基于活动目录（AD：Active Directory）、工作组、证书体系实现。对于活动目录和工作组方式，确保 Windows 7的宿主应用可以融入 Windows平台的运行环境；对于证书方式则可以将 Windows 7开发环境融入企业的异构 IT环境中，实现与其他 LDAP来源系统（例如：Unix、Linux、ORACLE和IBM商用套件等）的集成。

除了.NET支持外，Windows 7还提供对 PHP的集成，并通过 IIS 7中微软专门优化过的处理引擎和缓存机制实现PHP宿主，并通过 PHP封装 SOAP实现Windows 7上的PHP服务组件。

总体上，Windows 7的分布式组件和容器体系如图4。

消息机制

作为新的服务平台客户端，Windows 7除了提供对实时交互的支持，还针对网络和交付要求提供增强的异步消息支持，相关处理主要借助 MSMQ完成。Windows 7内置的msmq是 V 5.0，增强如下：

- 首先，选择 SHA-2为默认的散列算法。这对于之前版本的 MSMQ存在一定兼容问题，因为之前版本默认用SHA-1，为了实现新旧 MSMQ的通信，需要配置Windows 7的 MSMQ注册表允许 WeakHashAlgorithms。不过打开这个选项也意味着提供对 MD2、MD4、MD5的支持。
- 去除了对Windows 2000客户端的支持服务。由于原有的服务依赖较早版本的Domain Controller负责消息协议，为了保持和Windows 2008 R2+ Domain Contorller的协同，此次去掉了一个技术大包袱。对于使用新Windows平台的维护人员而言，可以免去同时维护新旧环境的问题，尤其对于一些位于 site边界DMZ区域外的服务器，使用Windows 2008 Server Core的小attack surface（攻击面）模式产品也许是个不错的选择。



图4 Windows 7平台分布式组件和组件容器

- 提升众多队列情况下报文的加载效率，通过改进查找算法，即便在Windows 7定义数千个Queue，也可以获得不错的启动效率。随着offline和偶联计算的普及、云计算的推广，未来Windows 7作为一个计算客户端平台可能要面临越来越多的异步连接要求，容量的扩展对未来企业计算颇有意义。

不过，作为一个企业级客户端队列环境，Windows 7还有很多不足，必须借助中间件完成高级的消息机制，可能的改进领域包括：

- 借助非事务性队列提供事务的访问控制。虽然听上去有些悖论，但现实中却经常会需要类似的特征，毕竟企业环境中MSMQ交互的对象往往并非建立在可靠的交付通道上。除了发送 /接收消息外，还必须增加一系列控制实务性的消息交换，但这些控制并非总是有效，上层中间件为了简化业务功能实现又常常假设有效。因此，是否能够在 MSMQ的未来版本中提供瘦身后的事务交换，并在实现中增加必要的补偿机制是很多Windows平台应用架构师所关注的；
- 报文的 4M容量限制也亟待改进，毕竟现实的应用不总是“hello world”，虽然通过 BizTalk的后台自动分包机制可以传输上 G的数据，但微软似乎还不愿意在MSMQ上作出改进。

国际化（I18N）和本地化（L10N）

在被互联网拉平的世界中，您的用户在不久的将来也许会在世界的任何角落使用 Windows 7，根据市场情况您需要确保自己的应用能适用于不同的语言及地区。国际化（I18N：internationalization）是指在设计软件、将软件与特定语言及地区脱钩的过程。当软件被移植到不同的语言及地区时，软件本身不用做内部工程上的改变或修正。本地化（L10N：localization）则是指当移植软件时，加上与特定区域设置有关的信息和翻译文件的过程。国际化和本地化方面 Windows 7则基本延续Windows vista的内容，毕竟 Windows从Xp版本开始已经是个I18N ready的产品，至于如何L10N则一直在改进。

从企业应用架构的角度看，Windows 7的 I18N和 L10N的一个潜在问题在于其标准的选择上。毕竟，企业环境往往不仅 .NET平台一家，以 Java EE 5/6为例，区域（Locale）对象遵循的是三段方式：

Language（ISO 639）+ Regional Variation（ISO 3166）+ Variation

而 Windows .NET平台主要采用 3字母的方式表示一个区域，采用的技术标准也略有差别，但企业应用架构中即便通过 SOA环境可以提供消息互联，但 L10N的异构平台应用同时用于Windows 7时，就可能产生冲突。为了避免上述问题，架构师可能需在 .NET、Java EE应用的 I18N和 L10N上增加一层抽象，确保 Windows 7客户端运行的web 应用和 Desktop应用都能正确的服务用户。

安全性

Windows 7安全性方面的增强倒是非常可圈可点。

随着信息安全概念的不断外延，提到information security，更多的集中在管理和管控方面，但对于Windows 7这样一个产品而言，企业应用架构中它位于IT环境的底端、位于IT环境的边界、甚至位于IT环境外部。但同时，它也是供给者借助各种应用（Web / Desktop）渗透进企业IT环境的主要通道。那么Windows 7配合Windows server 2008 R2的管控手段有哪些新措施能帮助我们更小的缩小attack surface呢？

安全性和易用性始终是个矛盾，做多了用户烦、少了用户怨。Window 7则把这种控制简单到一个拖动，这样即便是普通用户也可以根据工作环境安全防护的差异设置来保护自己。这个功能虽小，但不用手工改注册表、改本地策略等，简单、实用，有点上善若水的味道。

Direct Access也具有类似的效果，它可以视为对 VPN的有益补充。它使用IPsec进行计算机之间的认证，管理员可以在用户实际接触后端服务器前增加额外的策略控制，可以说把IP 6的特性发挥得很到位。而且当IpSec不可用的情况下，Direct Access甚至能通过HTTPS协议完成通信（尽管效率稍慢点）。该特性的引入，可能对未来企业内部远程管理的设计方式有所改变。

AppLocker和 Biometric Framework前者作为新的组策略功能，可以有效限制脚本、程序的执行，能够比较好的防范普通用户意外信息泄露。后者则是对生物识别技术的统一，以往类似技术往往被不同厂商标准所垄断，API体系也林林总总，而这个framework可以作为未来 Windows平台开发、集成相关生物安全技术的参考标准。

上面几个新特性只是Windows 7安全增强的一部分，但却提示我们不要总期望通过配置文件和自动更新解决普通用户面临的安全问题，终究他们不是专家也不是应用的设计者，好的default settings + 简单易用的配置工具也许更有实际意义。软件以人为本在客户端安全方面更是如此。

总结

上面是从架构设计的 9个方面对Windows 7进行考察，不过作为架构师您的工作就是“穿针引线”，把 Windows 7的众多设计思想和特性与您的应用缝在一起，借助增强的Desktop控件和 IE 8把他们更好的呈献给最终用户。



作者简介：
王翔，软件架构师，主要研究方向为XML、.NET、领域设计和PKI应用。工作之余喜爱旅游、写作和烹饪。

客户端与服务器的对话

——Windows 7 是新产品吗？

■ 记者 / 白苗

与Windows Vista推出不同的是，曾经开发代号为Longhorn的Windows第六代产品发布之前，不仅大张旗鼓地对客户端技术进行了更新，更重要的是，在服务器端也同样进行了大规模的架构革命。然而，当Windows 7被微软孤零零地扔出来的时候，我们不禁要问，Windows 7没有服务器端吗？



微软公司服务器与开发工具副总裁Robert Wahbe证实Windows 7对应的服务器版本就是Windows Server 2008 R2。

没有Windows 7 Server

这道题的答案是全球副总裁Robert Wahbe在日前的一场媒体见面会上带给我们的："Windows 7用户可以与所有服务器产品做互动，不需要有特别的服务器版本来支持。相对应的产品有Windows Server 2008。其实，Windows Server 2008产品已经包含了众多重要的技术，比如DirectAccess、BranchCache等，都完全与Windows 7和Windows服务器结合。"

这意味着我们不会看到Windows 7的服务器端产品。仅从客户端角度看，Windows 7在架构上与Windows Vista并没有太大的差别，尽管加入了不少新的功能，但是其内核以及底层平台却与上一代产品保持了高度的一致。不同之处除了加入了一些更友好的特性外，就是在性能和兼容性上有了较大的提升。

从这个意义上看，Windows 7更像是Windows 6.5，而根据维基百科提供的资料显示，Windows 7的操作系统内核是Windows NT 6.1。在对一些业界专家的采访时，甚至有人认为Windows Vista看起来就像是Windows 7的Beta版，这无疑让我们感觉，Windows 7并不是一个新产品，而是一个姗姗来迟、让我们满怀期待的新操作系统。

Win32还是.NET

对于开发者而言，最关注的仍然是Win32平台与.NET平台的问题。自从Windows Vista问世以来，微软的战略就越来越多偏向.NET托管代码平台，而原来的Win32平台开发者，尤其是像Delphi这种基于Win32原生平台进行开发的程序员，对于未来是否需要转向.NET平台忧心忡忡。

对此，Robert Wahbe的解释是："我们针对的客户从终端消费者到企业级用户，每个人的需求都有所不同，有一些是以互联网应用为基准，他需要的平台技术和开发工具，也有相对应的针对性，有一些则需要开发性能更好的应用。未来，我们会继续保持在Win32平台上的投资，同时给Win32平台的开发者和.NET的开发者提供充分支持。"

因此，微软未来的服务器技术和开发工具领域技术不会发生太大的改变。尤其是在新的虚拟机技术以及数据库技术的进步下，原有的系统级软件开发人员仍然有很大的发展空间。

小结

随着Windows 7推出，服务器端的技术相对保持了较高的稳定性。未来除了客户端技术的更新之外，服务器端的发展方向必然是走向虚拟化。以Hyper-V为代表的虚拟化技术，更是值得开发人员关注。

比如美国大陆航空公司（Continental Airlines）的开发人员表示："通过使用Hyper-V技术和远程桌面服务，Continental能够最大化系统运作时间，数分钟内即可部署新的服务，无须等待几个星期。在软硬件、人力、实施以及相关运营方面，Hyper-V技术节省的开支，一年可达150万美元。"

这同时也表达了另外一层意思，技术的变迁尽管很快，但是作为程序员，应该多花一些时间了解技术所实现的业务这一本质问题。即使Windows 7带给开发人员的改变并不多，但是我们应该始终关注这些产品和技术，能否给我们带来真正意义的价值。

嵌入式，与 Windows 7 保持同步

■ 记者 / 白苗

作为Windows的另一个分支，Windows Embedded一直以来就与 Windows的发展保持着同步。10月13日，也就是在Windows 7桌面操作系统正式发布前10天，微软向市场发布了新一代的Windows Embedded Enterprise和Windows Embedded Server平台的RTM（release to manufacturing）版本。微软公司 Windows Embedded中国区高级市场经理李涛先生介绍：“Windows Embedded Enterprise的系列产品全面采用了Windows 7的技术，而其Windows Embedded Server版本则采用了Windows Server 2008 R2作为基础平台。”



微软公司Windows Embedded中国区高级市场经理李涛说：“Windows Embedded Enterprise系列产品都基于Windows 7的全新技术。”

嵌入式的新技术应用

从功能上看，作为Windows 7亮点的主要是多点触摸技术和企业级连接技术。这两个技术在Windows Embedded Enterprise系列产品上得到了很好的体现。由于嵌入式软件通常都安装在专用设备上，以信息亭和 ATM等应用服务终端为代表，这些设备通常都需要使用多点触摸的控制功能，新技术的出现给人们操作这些终端提供了更大的便利。

另一方面，一些便携设备，例如手机、信息采集器等移动终端通常需要强大的连接功能，在已经报道的Windows 7技术中，DirectAccess和BranchCache技术则帮助移动终端可以通过非常便捷的方式访问企业内部网络，而不需要再搭建成本高昂的虚拟专用网。

此外，由于 Windows 7本身的兼容性是微软此次发布时的一个重点。桌面端系统的兼容性测试，对微软开发团队的要求是100%兼容 Windows Vista。因此，在嵌入式领域，新产品要求对原有Windows XP Embedded系统上运行的应用程序也达到 100%的兼容。不同的是，在Windows Embedded Enterprise系列产品的两个版本（Windows 7 Professional for Embedded Systems和 Windows 7 Ultimate for Embedded Systems）当中，都有 Windows XP Mode提供的虚拟 Windows XP环境来达到这个目标。

服务器端变化不大

作为与 Windows 7配套的Windows Server 2008 R2，其嵌入式版本原先并没有相应产品。此次推出的嵌入式服务器版本，主要引入了 Windows Server 2008最初的很多特性，比如最小化的 Windows Server Core技术、Hyper-V虚拟化技术以及最新的管理脚本语言PowerShell 2.0。

此外，由于 Windows Server的嵌入式版本需要与很多不同的专用设备进行交互，因此在可连接性和可靠性上进一步进行了加强。除了增加 Agile VPN等技术外，也集成了DirectAccess这样的连接技术。

未来产品和方向

发布会上，李涛对未来专用领域的市场进行了解释：“到2012年嵌入式设备（包括手机）市场会达到150亿美元的容量，微软看到了这个趋势，并将嵌入式领域的发展战略瞄准这个方向。在此期间，包括开发工具等相关的配套技术也会陆续出现。”

此外，微软还在前不久宣布了Windows Embedded Standard 2011 CTP（Community Technology Preview）版本。作为基于 Windows 7技术的组件化操作系统，这款产品将会于 2010年上半年面市。随着Windows 7技术的不断推广，除了PC平台外的嵌入式和服务器领域，Windows 7技术和产品也将逐步更新到新的设备上。

谁为 Windows 7 买单？

■ 记者 / 谭茂

带着无限的憧憬以及无数人的期许，Windows 7来到了我们的眼前。可以说，Windows 7是集万千宠爱于一身，有人说，它是微软挽回市场的“撒手锏”，还有人甚至将Windows 7赋予了IT产业在经历了一年经济寒冬之后的复兴大任。

微软大中华区Windows商用产品部总经理Adam Anger日前在接受采访时也自信地谈道，“与以前发布的Windows 产品版本比较，Windows 7在市场上的需求是最大的。”

但是，Windows 7真的能如人所愿吗挽救市场、挽救微软吗？面对微软号称所谓的全球“最低价”操作系统，到底又有多少行业客户会为它买单呢？

微软的Windows 7算盘

眼下各大媒体机构都在预测 Windows 7的前景，虽然大家的看法大相径庭，但有一点大家的看法是一致的，Windows 7必须避免走上 Vista上的老路，此外，如何让用户尽快抛弃熟悉的 XP，以及在Mac、Linux等微软的竞争对手面前显示出自己优势，这是Windows 7要想取得成功的关键。

从某种意义上来说，Windows 7对微软是一个只能成功，不容失败的产品，因为在Windows 7上，微软已经低下了高贵的头，认识到了自己所犯的错误并从中吸取了教训，且动员了一切可动员的力量，其所投入的资源也超过了历史上任何一个操作系统。在生态改造方面，微软花了好几个月时间，与合作伙伴、终端用户进行密切的配合，力争使整个产业链做到Windows 7快速就绪。

微软大中华区Windows商用产品部总经理Adam Anger日前在一次面向媒体的Windows 7说明会上表示，微软已经在Windows 7兼容性上做了大量的投资。目前在市场常用的硬件中已经实现92%以上与Windows 7兼容，在软件方面Windows 7和市面上91%的软件兼容。

可以说，这个结果大大高于了微软的上一代操作系统——Vista。实际上，微软对于Vista的宣传营销手法，曾一度吸引了大量用户，但是由于其兼容性、安全性、烦琐性等一系列缺点遭到了用户的抛弃。

Adam Anger认为，Windows 7之所以值得注意，是因为在生态系统、市场开放方面，微软在Windows 7上的做法和Vista确实不一样，并同时改善了许多Vista的缺点。然而，微软花大量时间精力为Windows7造势，真的能够为微软带来实实在在的销量吗？来自用户的反馈也许才最有说服力。

光有兼容性是不够的

陈鹏来自于江苏省沛县财政局，对于Windows 7，他坦然，虽然一些财务软件比如新中大公共财务软件、财政票据管理软件等Windows 7尚无支持，但他仍旧打算在三个月内部署Windows 7。

根据调查结果显示，几乎所有行业用户都将兼容性放在了升级系统的第一考虑。最高的当属金融行业与教育行业，各自行业分别有55%与51.2%的用户认为兼容性是第一考虑因素。

升级成本问题是兼容性过后的第二考虑因素，从统计结果来看，医疗与制造行业对于成本问题最为看重，分别达到了34.8%与30.6%的认同率。西门子中国分公司IT运营总监刘辉在谈到Windows 7的关注点时表示，西门子部署Windows 7涉及的关注点，最主要的是成本、系统应用和兼容性问题、系统稳定性等特性。他表示，“希望Windows 7能够为西门子带来更多的有效性，以及高效的桌面应用、特别是兼容性和行业应用方面”。

实际上，对于升级成本上的问题，用户已经表示出了足够的关注。

对于成本的问题，Adam Anger这样解释道，“对于企业而言，Windows 7升级会带来巨大的回报。首先是Windows 7在生成镜像方面比以前系统少用50%时间；其次，Windows 7中已经内建方便企业客户进行PC管理的功能，以实现成本节约的目标；而通过PC管理功能，客户每年每台PC管理上成本

可节约70-160美元。”

而用户对Windows 7另一个较大的关注点来自于安全性。

上海良友公司是一家从事房地产的公司。对于他们来说非常重要的一点，就是保护自己PC机上的地产信息、客户信息。该公司技术总监在谈到对操作系统的需求时明确表示，公司所希望的Windows 7提供的价值是可以利用Windows 7上的所带的非常先进的文档加密工具来保护企业的关键信息。

但仍旧有用户对于Windows 7的表示出了消极的看法。某用户在接受电话采访时表示：“我认为，微软提供一个很好的产品，只是时机很糟。”他表示，尽管Windows 7有许多令人赞许之处，但不巧此刻企业缩减对新技术的投资，IT预算逐渐缩水。他说：“即使我们有意升级，也无法确定升级的时间。还有许多IT事项待办，而预算日益缩减。”

毋庸置疑地，无论是从性能、图形用户界面、兼容性、小工具还是安全性方面讲，Windows 7都是一款出色的操作系统，绝大多数用户对Windows 7表现出了足够的好感。但问题在于，有多少用户能够真正地为 Windows 7买单呢？ Windows 7销售最大的障碍又在哪儿呢？

Windows 7面前的三座大山

如果把 Vista比作是用户心目中一个阴影，那么XP在用户心目中更像是一个熟悉的爱人。从某种意义上而言，让用户抛弃XP的难度恐怕更大。

Adam Anger在其演讲中多次地对外界传达这样的信息——如果企业继续使用 XP，将会因支持老版操作系统的费用而增加成本。

在中国，XP还占据着绝对份额的市场，但随着2014年Windows XP的截止使用期限的到来，XP迟早将会被微软抛弃。实际上，而为了强迫中国人远离 XP，微软在一年前就开始了行动。如果说关掉洪磊的番茄花园只是微软借打击盗版切断XP的生命线，那么“黑屏”则是微软“教育”用户的一个重要手段。

在谈到盗版问题时，Adam Anger明确表示出了微软的态度，“对于Windows 7的反盗版策略，将采取与电脑厂商合作以及教育用户的两种方式。具体来说，微软将积极与 OEM厂商合作，做到新机预装的都是正版Windows 7；而另一方面，也将积极地教育用户，为正版用户提供增值服务，让用户知道正版的价值不同于盗版，让用户知道使用正版的安全性，以及其可以获得免费的升级补丁服务。”

在Windows 7的市场层面，微软打出了第一张牌——低价。

据相关信息显示，在售价方面，微软 Windows 7价格要比Vista更便宜，微软Windows 7中文版部分零售盒装产品的价格，其中价格最低的为 399元人民币的普通版，而高级版的建议零售价为人民币699元。对比之前Windows Vista高级版零售价为239.99美元，Windows Vista商业版零售价为 299.99美元以及Windows Vista旗舰版零售价为399.99美元的价格，应该说是优惠了不少。

低价或许是最有效的撒手锏，但值得注意的是，老版本用户如果要升级到Windows 7的话，多少还是要付出点银子的，因为可以免费升级的用户仅限于2009年6月以后购机的用户。



微软大中华区Windows 商用产品部总经理 Adam Anger 介绍Windows 7商业价值。

因此，除了低价，微软还针对企业用户推出了“Windows 7企业快速就绪计划”。Adam Anger表示，“所谓的快速就绪计划，实际上是指2010年2月 28日之前购买 Windows 7升级版批量许可的企业用户，将享受6.5折的优惠折扣。”

低价 +折扣，应该说，在经济形势不容乐观的今天，微软是充分考虑到了用户的接受能力。但在微软看来，这仍然不够。

“Windows 7能够在旧电脑上运行”——这是微软打出的第三张牌。

“过去企业需要购买新电脑才能够使用新版本的Windows操作系统”，Adam表示道。但是现在，微软正在极力劝说那些想升级系统但预算不足的CTO们，“买Windows 7吧，您甚至不需要升级硬件就能安装它。”

对微软的OEM合作伙伴来说，这也许并不是一个太令人振奋的消息，正因如此，相当多的硬件厂商开始担忧Windows 7的上市对年内PC销量的拉动作用微乎其微。就连鲍尔默日前也公开坦诚，“Windows 7不会给PC市场复苏带来很大动力。”

但不管怎么样，寥总是胜于无的。

小结

也许，企业升级至Windows 7只是个时间问题。在微软发布Vista之后，一些客户甚至放弃了购买PC的计划。Vista的失败和随后发生的全球性经济衰退，可能会有助于Windows 7取得成功。许多企业当前都面临着软件和计算机老化的问题，因此会有许多企业渴望着进行升级。一旦经济复苏后，Windows 7毕竟是有助于企业提高支出预算的。

Windows 7 研发背后不为人知的故事

■ 记者 / 付江

中国？还是印度？

本来仅仅打算走个过场，但微软总部考察团中国行的所见所闻，以及感受到的中国研发团队的实力和决心却最终让他们改变了最初的计划。眼前，对于考察团负责人Mohammed和James来说，当前最重要的任务就是对微软分布在全球各地的研发团队进行全面考察，再从中挑选出最适合承担软件兼容性测试任务的团队，来负责Windows 7中除了英语之外的22种国际语言的应用程序兼容性测试。由于将采用黑盒测试，加上考虑到传统的研发实力和人力成本因素，考察团将本次考察的重点放在了中国和印度这两个新兴市场。

毫无疑问，如果简单地从天时地利人和来看，印度在这场竞争中占据了一定先发优势。首先，印度有着比中国更加成熟和规范的大型IT外包团队；其次，Windows 7英文部分的软件兼容性测试已经放在了印度，如果把国际语言部分也放在印度，可以跟印度的英文测试团队密切配合；再者，相比中国，微软和印度以往类似国际项目合作更多、经验更丰富，总部或许更相信印度的研发实力。

2008年3月，微软总部考察团第一站来到了中国，当时主要关注的内容有两项：首先需要考察的是当地研发团队自身的业务能力和以往的工作成果；其次考虑到本次任务工作量的繁重，目标市场的整体产业生态环境也成了评估的重要指标。对于微软中国研发集团来说，早在2003年的时候就已经做了相应的国际化项目准备工作，当年做Windows 产品Debug时就启用了外包团队，积累了宝贵的与合作伙伴协同经验，当时也承担了一部分国际语言项目的测试工作。再加上本次项目组表现出了丝毫不逊于微软雷德蒙总部团队的研发能力和敬业精神，这些表现都给自己加了分。考察团对此很满意，随即与总部商量暂缓了对印度的考察，将重点转移到了中国。

两个月后，由测试经理Oliver带队的第二拨总部考察团再次来到中国，对中国研发集团的合作伙伴进行考察，最终选定文思创新公司。而文思公司胜就胜在准备充分上。由于本次项目的难点在于需要测试多达22种国际语言的应用程序，对于承包方来说最大的困难就是找到懂这些语种又精于测试的工程师并组建团队。对于文思来说，早在Windows 7项目之前，已经通过全球各国的分公司建立起了国际化的人才招聘渠道。而在Win 7项目竞标过程中，文思已经成功招聘了20多名来自不同国家的外籍员工，已经覆盖了项目需求语种的70%。在2008年5月底最终确定中标后，文思公司在接下来的一两个月内最终完成了这支“多国部队”的组建工作。

在考察过文思公司并向美国总部汇报沟通后，考察团最终取消了原定的印度之行，下定决心将项目留在中国。

软硬件兼容性决定Win 7成败

自2007年1月Vista上市后尤其在前期发布的版本中，其对于主流软硬件的兼容性差就一直为用户所诟病，其间包括腾讯QQ、网络游戏、部分网银网上密钥，甚至是微软自己的MSN都曾曝出过不兼容的问题，这些都对微软在全球推广Vista的工作增加了重重阻力。微软主管 Windows产品的全球副总裁Mike Nash也在访华的时候毫不讳言，称Vista在中国几乎就没怎么卖出去，不少用户在试用后非常失望继而继续降级使用XP操作系统。

正是由于过去对软硬件兼容性问题的忽视，导致了微软在Vista上打了“败仗”，微软高管在考虑Windows 7的兼容性时表现出了前所未有的重视。

过去在做Vista系统应用程序兼容性测试的时候，英文部分集中在美国总部做，其他语言由分散在世界各地的研发团队分别来做。这种模式对于中国这类研发力量较强的国家还好说，但欧洲的一些小语种国家

和南美地区，他们有自己的应用程序厂商，但缺乏足够的测试力量，导致最终产品兼容性也比较差。另外，由于缺乏一套统一的流程和标准，面对市面上数以十万计的应用程序，如何按统一的标准进行分类并抽样测试等问题也没有获得很好的解决，都成了Vista前期兼容性差的重要原因之一。

据微软中国研发集团 Windows国际产品组负责人丁泳介绍，软件兼容性问题一般主要考虑两个方面：操作系统本身和被测试软件。过去，例如在Vista及Windows前期家族成员的规划开发中，应用程序兼容性按惯例都是放在开发后期才加入考虑，对不兼容的ISV应用程序，处理的方式大多是让ISV自己去修改应用程序来主动适应Windows。而对于Windows 7，微软总部负责设计产品功能的Feature team从最初设计的时候就充分考虑了软硬件的兼容问题。对待类似的Bug的态度，微软自身也变得更加主动。Windows International团队对每一个出现的问题都会深究根源，如果问题是由操作系统自身造成的，则会与微软总部沟通，更改系统程序，使问题得到根本解决，而不再是一味的要求软件开发商更改其程序。

丁泳举例，在过去ISV会假定微软Windows的API是这样的，这种假定在XP或Vista的环境下是可以正常运行的，但当操作系统升级到Windows 7后，这种假定的条件可能会发生改变导致该应用程序无法在新系统下运行。通过与ISV协商，微软甚至可以主动提供一个Windows 7系统本身并不必需的特殊“钩子”，让应用程序在这个虚拟环境下继续运行，由此来主动适应一些普及率很广的应用程序。让微软来降低“身份”适应ISV，这在过去很难想象。

微软方面对兼容性考虑的目标有两个：(1)过去在 Vista系统上能运行的应用程序 100%都能在Windows 7上运行；(2)在Vista上运行不了，但在XP系统上能运行的主流运行程序尽量也能在Windows 7上跑。

RC版发布前惊心动魄15天

Windows7 RC版本发布的最后时刻，中国研发集团与美国同事们还经历了惊心动魄的半个月。事情源于发布前的最后一个Bug。

距离Windows 7 RC版发布还有15天。2009年4月5日，微软总部Feature team对Windows 7 Kernal32.dll再次做了优化。4月6日起，陆续从不同渠道报告有多个应用程序不能在最新的Windows 7 Build上运行，并报告“An internal exception occurred (Address: 0x0) ”错误，其中就包括多款韩国在线网游如天堂(Linage 1)。4月8日，中国研发团队开始调查。4月14日，中国研发团队通过艰苦追踪工作基本明确了问题所在，源起应用程序依赖于Kernal32.dll的导出表，开始与总部Feature team沟通。中国团队及时准确的调试调查工作，是保证Bug能得到及时处理的关键。



微软中国研发集团Windows国际产品组负责人丁泳带领团队将除英语外的Windows 7软件测试工作全部接下来了。

4月15日/16日，总部Feature team进一步调查确认了该Bug。此时可供选择的解决方案有两种：第一，忽略该Bug，按原定时间发布RC版本，之后在下一个版本(RTM)或过程中打补丁。因为当时距离RC版发布已经不到一周时间，而通过评估，如果要对该Bug进行修复，需要对Windows 7内核代码进行修改。而对内核的修改往往“牵一发而动全身”，工作量和风险都较大。第二种解决方式，不惜一切代价立即解决掉该Bug，按时发布。

最后 Feature team还是在4月16日提供了建议采用第二种方式的解决报告。4月17日，Fix测试通过，并经过总部 (PM/Dev/Test)三部门负责人签字和ShipRoom（相当于总参谋部）的批准，正式决定限时解决该Bug。

在这段时间里中国方面一直在对Bug进行监控和处理，得到修改的决定后马上对新的修改进行全面的测试，以保证不会有新的问题出现。4月18日当天发布的版本中该Bug问题已经得到解决。

4月21日 Windows 7 RC版正式如期对外发布。

编后语

对软硬件兼容性的定位，对待Bug的处理态度都可以发现如今微软在Windows定位上的转变，作为IT业界的航空母舰，微软也改变了过去“老子”一家独大对ISV爱理不理的态势，从过去让业界生态环境来适应自己，在产品升级的时候主要以Feature-drive，到如今Windows 7规划开发时主动适应ISV，把兼容性当作战略级任务来抓。从过去以每个新版本增添多少新功能为首要目标到现在更加聚焦整个生态环境的协调发展，不得不说都是微软做出的重要调整。

编者按：英特尔一直是微软长期以来最密切的合作伙伴。然而从去年开始，英特尔就高调宣布了其基于Linux的开源操作系统Moblin，并从底层硬件平台上定义了这个系统的适用范围：MID。然而，至少在上网本领域，Windows与Moblin是存在重叠的。那么这种竞争是否会演变成为两个亲密合作伙伴的战争？为此我们采访了英特尔亚太研发有限公司总经理梁兆柱博士。

Moblin 不与 Windows 7 竞争

■ 记者 / 白苗

Moblin所定义的终端平台被称为MID，从广义上说，比PC更大的计算终端不在这个范围内之外，其他的便携式设备全部可以称为MID。这是很多人对MID最早的印象。然而最近两年极为火的上网本，除了采用 PC 架构外，也是MID设备的一种，这让我们不禁生疑，难道英特尔要转行做操作系统？

三个MID的要素

为了看清楚这个问题，首先要看MID本身的定义。对这一点，梁兆柱博士并没有直接回答我们的问题，他用了另外一种方法来解答，并总结了三点：“过去十年最重要的技术变革是互联网，它出现的时候是一个信息交换平台。人们在互联网上交换各种各样不同的信息。今天却不一样，可能你在一个文本框里输入几个字符，在不到一秒钟之内，互联网就返回给你数千万的检索结果。毫无疑问，它变成了一个具有计算能力的平台。”

“更小的平台也是一样。这几年一个概念被谈得很多，那就是‘物联网’，这些车载设备、手持终端如何联接交互？他们必定需要计算能力。因此，具备计算能力的设备就不再仅限于PC，而是各种各样不同的设备，拥有计算能力，并可以进行交互。今天看到的手机购物就是一种典型。更重要的是，这种具有计算能力的平台基本上都用来提供服务，因此是服务性终端。”梁博士补充道。

“第三点，处理器变得越来越便宜，每片处理器的计算能力越来越强大，功耗也越来越小，原来那些无法安装处理器的设备，现在也可以具备计算能力了。比如一辆汽车，上面就有好几个处理器，甚至连气囊都拥有计算能力，能够根据驾驶者的状态，采用不同的力量进行充气，从而不会因为保护而伤害驾驶者。”

毫无疑问，那些对计算能力需求很小，计算场景也非常单一的环境下，人们并不需要一个像Windows 7一样的系统来运行。更轻量级（根据不同场景的需要而定制）、价格更便宜（最好是免费）、更规范化（接口和数据交换标准统一）的操作系统有很大需求，这是英特尔定义MID的核心。

软件还是硬件？

“如果我们看过去几十年时间就知道，比如1982年，那个时候的PC计算能力还非常弱，根本无法满足后来家用计算的需求。程序员想尽办法让计算机变得更智能，计算能力更好，就需要更优化的代码。”梁博士回忆道。

“今天，我们看到很多企业在强大的硬件上开发虚拟化技术，其目标也同样是希望将硬件能力充分发挥出来。而与硬件结合最紧密的操作系统，它代表了软件与硬件共同的发展方向。”梁博士说，“因此，光有硬件不行，光有软件也不行。”

这番话表达的意思在于，今天很多设备需要具备一定的计算能力，但如果有芯片（处理器）却没有软件运行，这个设备无法智能地提供计算，反之亦然。Windows 7所解决的问题主要是在PC上，它能给用户带来非常好的用户体验，却很难用这种软件去控

制一些日常我们看到的智能设备。对此，梁博士给我们举了个例子：

“我们经常可以看到路边的标识牌，这些标识牌上有很多需要不断更新的信息，比如道路指示、广告灯，这些信息明显无法采用手动模式进行更新。因此它一定需要拥有计算能力。”换句话说，这种不采用标准PC架构的“计算机”，根本不需要安装一个Windows 7，它只需要一个非常简单的操作系统进行控制即可。近乎没有成本费用的Linux计算内核无疑是一个很好的选择。

除了日常生活中看到的设备之外，涉及工业和商业的众多领域，都有许多设备开始拥有计算需求。英特尔可以根据这些设备的需求设计出处理器，但是却没有可以运行在这些处理器上的软件，这是英特尔为什么要做Moblin平台最大的原因——英特尔是一家处理器厂商，他要做的就是让这些智能设备可以运行在英特尔的处理器（比如Atom）上。

做平台不做产品

其实，Moblin是一个操作系统平台，它并不是一个完整的操作系统产品。真正的产品提供者，是那些采用了Moblin平台进行集成的OSV（操作系统提供商，Operating System Vendor）。

这也很好地解释了在上网本或手机领域，为什么会让我们感觉到Moblin在和微软的Windows进行竞争的现象：由于 Moblin本身具备很强的计算能力，能够完成PC或智能手机所需要的计算，因此它可以用来安装在PC和智能手机上，但这并非Moblin的历史使命，因为在这个领域里已经形成了完整的市场格局，只不过因为最近两年，上网本与智能手机市场是热门，而那些OSV厂商则采用这种更廉价的系统获取市场份额的手段。

“如果从整个计算机架构的角度看，我们就知道：运行在整个系统栈最底层的是硬件平台，与硬件平台交互的是操作系统平台，而在操作系统平台之上，则是软件的运行时环境。这个部分都可以称之为平台。”梁博士表示，作为处理器厂商，最了解如何最大化处理器的性能，因此英特尔与微软等操作系统厂商的合作将会持续进行下去。

Moblin作为一个开源的平台项目，同样是英特尔提出的“平台更多选择”。真正的软件运行时环境，英特尔似乎并不打算插手太多。

此外，今天操作系统的概念也被逐渐模糊化，事实上，Windows 7包含除传统意义上的操作系统，还提供一个软件运行时环境（.NET），这些技术通常都是在OSV对系统进行定制的时候选择的内容。对于掌握着Moblin的英特尔而言，开发商到底采用Java的运行时环境，还是采用本地代码编写程序显得并不那么重要。

产业生态链的建立



英特尔亚太研发有限公司总经理梁兆柱博士表示：英特尔的战略非常明确，而Moblin也绝不是Windows的竞争对手。

最困难的地方在于建立一个完整的产业生态链。Moblin的最终目标是实现各种设备上的处理器都是 Intel inside，尤其是针对其轻量级计算的 Atom处理器，因此，在硬件平台上进行了一定的统一是Moblin的先天优势。以开放的Linux内核作为 Moblin的基础更是锦上添花。

然而，有很多人大谈Linux的跨平台特性，但背后现实的情况是Ubuntu平台上开发的应用程序依然无法运行在红旗的系统上。应用程序的兼容，并不是操作系统内核统一就可以无条件实现的，这与大量软件程序库、外部设备等很多条件有关。

而工业和商业领域的各种嵌入式设备本身就存在很多不同的规格和标准，这项工作的统一显得尤其困难。梁博士介绍说：“通过与17家OSV合作伙伴的协商，整个 Moblin合作伙伴之内的厂商都已经达成了共识，一旦通过了Moblin的兼容性测试，应用程序将可以实现真正意义上的跨平台。当然，真正与外部设备等相关的兼容性，未来还有很长一段路要走。”

在这方面，梁博士还提到了日本的方式非常值得学习。日本的产经联在整个产业生态链的建立上非常完善，由大财团所控制的企业在面向全球市场的时候，始终保持较高的统一性，从而达到有效兼容的产业保护状况，这种理念也将为Moblin建立产业生态链提供参考。

小结

通过对Moblin战略的了解，我们很清楚地看到，英特尔与微软的联盟关系依然稳固地存在，只要未来的Windows依然在X86的英特尔架构上运行，这两者之间依然不会存在竞争关系，因为他们的战略早已经注定，英特尔做处理器，微软做操作系统。

编者按：从Borland到CodeGear，再到英巴卡迪诺，不管名字怎么变，其开发人员的特质总是不会被人们忘记。尤其是Delphi，这个始终与Windows开发站在同一战线上的微软的朋友，在面对Windows 7出现的时候，会表现出怎样的目光？《程序员》特别采访了英巴卡迪诺大中华区总经理琚伟。

Windows 7 不是微软的革命

——英巴卡迪诺大中华区总经理琚伟访谈

■ 记者 / 白苗

作为一家开发工具厂商，Borland算得上是最了解开发者的企业。尤其是随着Windows客户端操作系统的崛起，大量程序员曾经使用运行在微软Windows平台上的Borland开发工具，编写出无数激动人心的软件。尽管微软在Vista时代之后，将开发平台进一步迁移到 .NET平台上，但Borland与微软之间千丝万缕的联系就从来没有切断过。

和微软一起做开发

“即使在今天，微软的核心开发人员仍然有不少是来自Borland公司的。这当中甚至包括 C#之父 Anders Hejlsberg。因此，如果说 Windows 底层平台有任何新的进展和变化，Borland都应该是知道得最早的一家厂商。”琚伟在采访一开始就提到这个重要问题。即使是在易主英巴卡迪诺以后，这家公司的工程师仍然与微软的老同事们保持着亲密接触。

作为在软件开发领域微软最紧密的合作伙伴，Delphi这个开发工具总是能够体现微软开发平台最新的进展。无论是底层Windows API的变化，还是 .NET托管代码平台的变化，Delphi产品总是会先人一步将这些新的特性反映在其产品当中。由于Delphi几乎每年都会有一个新版本发布，因此其更新速度甚至超过了3年才发布一次的Visual Studio。换句话说，最了解开发者和最了解微软的，都是Delphi。

“应该说，Windows 7在用户体验上，尤其是消费类用户体验上有了一个很大的提升。然而，这并不代表Windows 7是微软历史上具有革命性的产品。”在琚伟看来，Windows 7并不像众多媒体和博客吹嘘的那样完美。这是因为，Windows 7对开发者而言，所做的变化实在少得可怜。从过去Windows的几个版本看，无论Windows XP、Windows Vista，还是即将发布的Windows 7，Delphi都是最早支持在这个平台上进行开发的工具。

用琚伟的话说，微软需要有一个像英巴卡迪诺这样的合作伙伴。这不仅对微软资深有利，而且即使到今天，Delphi广大的开发者用户仍然是微软希望拥有的。这也让微软和英巴卡迪诺在很早之前就签署了协议。我们看到，Windows 7发布前，Delphi 2010的开发工具就已经面市了，其新特性不仅支持最新的 .NET框架开发，还支持很多Windows 7中加入的特性，如多点触摸技术等。

对于开发者而言，最关心的问题莫过于Win32平台和 .NET平台的关系问题。琚伟认为：“尽管从Vista时代开始，微软就试图将大量的开发工作都转交到 .NET平台上，但是Win32平台仍然会有相当长的生存周期。”截止到目前，Delphi仍然是开发人员做 RAD开发最好的工具，用户没有道理放弃Win32平台。另外一方面，微软也会对此进行一定的平衡。毕竟有太多的应用程序师构建在Win32体系上的。这些应用除了包括大量的桌面应用外，还有很多行业应用，由于可以保持非常良好的工作，行业开发者根本不用关心是否需要迁移到.NET平台上。

即使是这样，Delphi的战略也在于微软保持同步开发的同时，开始引入更多新的技术。最明显的例子就是PHP开发。同时，我们今天通过Delphi，也开始发现跨平台的特性，正越来越多的出现在Delphi工具里。琚伟也表示：“在未来的Delphi版本中，除了微软的技术，我们也将同时支持更多逐渐成为主流的技术，并强化到Delphi的新版本中。”

微软的挑战者

其实从某种意义上看，Windows出现以来，微软从来没有在客户端操作系统层面上遇到过任何像样的对手。Linux也好，Macintosh也罢，这些狂热的技术派和优雅的少数派从来都不是Windows在市场上的对手。人们常说，微软的对手只有他自己，但是在新的环境下，却也未必就像这些乐天派所认为的一样。

由于Windows的成功来自PC的成功，这座基石如果发生了动摇，微软的地位也将变得不再那么重要。今天，我们看到越来越多的平台，正在对传统的PC体系发起激烈的冲击。其中最有力的打击来自互联网和移动终端。这些都是微软在传统模式下不得不面对的强大对手。

Web平台正在逐渐转化人们使用电脑的习惯。对今天不少用户而言，用 Windows+IE访问互联网或是用Linux+Firefox访问互联网，并不是很大的区别，而以浏览器插件出现的RIA平台，更是给Web带来了极为丰富的用户体验。换句话说，人们对Windows的依赖越来越小，这本身也代表了微软所面临的严峻挑战。加上以 Google为核心的 Web厂商凶狠地抢夺市场，微软想要通过Windows在这一领域继续保持垄断地位，已经远远不像当初那么容易。

另一方面，来自移动终端平台的挑战同样不可忽视。今天在全世界保持运行状态的手机已经超过 20亿部，直接可以与PC一争高下。更重要的是，手机的计算能力越来越强，也越来越贴近用户。它给手机用户带来的体验提升以及影响力开始逐渐超过PC。“尤其是在运营商的介入之下，系统与手机用户之间的距离越来越近，同样对PC领域有着很大影响力。”琚伟说，“比如最近在中国市场上的Ophone，就很值得看好，恰恰就是因为运营商距离用户比系统厂商更靠近。”

反观微软，“新的 Windows 7从我的角度看，并没有什么理念上的变化，不管是技术理念还是市场战略理念，都和原来比尔·盖茨的时代如出一辙，这也是我为什么说Windows 7并不是革命性产品的重要原因。”也就是说，在面对开源、移动领域的穷追猛打之后，“Windows 7作为更新换代的产品可能还会延续XP的风光，但微软确实需要一场革命性的变化了。”

其实，早在很多企业都意识到互联网等新平台崛起所带来价值的同时，微软也看到了这个趋势。云计算也好，SaaS也好，RIA也好，微软都在不遗余力地参与，包括 Windows Azure、Silverlight、Dynamic CRM等产品和平台，都是微软投入了大量资源的项目。然而，过于优秀的历史，在进行战略调整的时候成了微软的负担。从某种意义上说，微软提出的S+S是一个不伦不类的战略，因为它既不清晰，也很难执行。

“对于新平台，我很看好Android和Mac平台，这些具有活力的平台在不断坚持给用户带来很多不同的东西，



英巴卡迪诺大中华区总经理琚伟认为Windows 7对微软来说根本不是革命性产品。

它们充满了个性，并且特点突出。”琚伟在对未来操作系统的发展上这样认为，“当然，这也并不是表示Windows 7不能重现Windows XP的辉煌，不过全面升级到Windows 7将很可能是一个漫长的过程，结果最终将会由市场的考验让各自交出答案。”

关注应该关注的问题

Windows 7发布在即，作为开发者毫无疑问也会非常关注。关于开发，Windows 7却没有太多值得关注的问题，尤其是相对此前的几代产品而言。因此琚伟认为：“作为开发者，应该尽可能关注自己所作的工作本身，比如程序员手中的项目。”

“今天我们看到了越来越明显的趋势，亚太地区的程序员越来越多，已经开始超过欧美地区。这样一个庞大的开发团队，应该如何有效提高自己的附加值，让我们所编写的代码更有价值。”这种更有价值的工作，需要软件开发人员更加专业化。这里有两点需要尤其注意：

提高工具的作用，并加强协作化开发。数量众多的程序员，如果不经组织，开发效率会大打折扣。“一方面，工具本身可以整合很多不同领域的知识，并通过规范化的流程加强程序员之间的合作，另一方面，也能够有效减少生产力的浪费。今天我们所作的软件项目，成功率只有30%，并不是因为开发能力不足，而是因为浪费太过严重。”琚伟认为。而工具能够有效减少浪费，同时给工具市场带来的附加值，将会进一步提高整个产业的附加值，让大家在更高的水平上竞争。

加强开发人员的专业化。比如，Delphi程序员，除了应该了解快速开发的过程，还应该对数据库的开发有深入了解。在拥有了软件开发技能的广度基础上，一旦能够更加深入、专业化地做软件，将会取得更好的成绩。琚伟玩笑道：“俗话说：‘不怕千招会，就怕一招毁’就是这个道理。收购了Borland的英巴卡迪诺，将整个开发工具产品线打通，帮助开发人员进一步加强专业化，将会是未来很多程序员的选择。”

小结

“今天我们可以看到，IT的增长速度并不如很多机构预测的那样快，对于很多企业和产品来说，需要一段时间的沉淀和积累。”琚伟说，“Windows 7的发布值得庆祝，尤其是在前一代产品非常让人失望的前提下。作为程序员，应该弄清楚做软件开发的核心基本规律是什么，一旦把握了发展的规律，平台的发展和变迁，将不再是我们需要担心的问题。”

Windows 7 的推出与操作系统的变迁

■ 文 / 何伟平



作者简介：

何伟平，Postgre-SQL中文版文档维护者，《Perl 编程》第三版译者，Linux集群管理员及数据库研究人员和软件开发人员。有近十年的软件开发经验，在金融、工业和互联网等行业有着较长的经历。现供职于某互联网公司，从事大型集群管理以及技术指导方面的工作。

大公司都善于利用媒体作势，跨国大公司更是如此。微软就是这样的一个大公司。最近一段在IT新闻里面看到很多Windows 7相关的新闻，有的说，Windows 7出来提前终止Vista，有的说 Vista是微软最失败的产品。还有的顺带手的把 Windows XP、Windows Vista、Windows 7拉在一起做个大比拼。在这些宣传和文字里，经常能看到Windows 7全面胜出、Windows XP 是微软的最后的辉煌这样的字眼。

从微软构想说起

看到这些新闻和视频，不由得想起自己十几年来经历过的微软的操作系统，DOS和WIN 3.1就不说了，从Windows 95开始，微软在推广的时候做的广告和宣传都不能不说是精品，不过，操作系统？一个有点遥远的名词。操作系统？我得承认，做为一个开源爱好者，实在是已经忘记了桌面操作系统的事情，长久以来使用着被称作异类的Slackware/Linux，只有在就职的公司摆放的机器上才运行有微软的操作系统，时间长了真的有些淡忘操作系统这档子事儿了。

近30年以来，操作系统是整个PC体系的IT产业的基石，微软自从DOS时代开始傍上IBM PC之后，越做越大，已经基本上做到了一手遮天的位置。当年的大哥IBM早已不被微软放在心上，随着DOS、Windows 3.1、Windows 95、Windows 2000、Windows XP到现代的 Windows Vista和Windows 7这一代代的操作系统的推出，微软一直牢牢占据着桌面操作系统的绝对领导地位，并且在这短短的二十多年里，微软利用其桌面操作系统的先天之优，战胜了一个又一个曾经尝试在桌面领域占据一席之地的对手，从办公套件到开发工具；从3D引擎到浏览器，现在看看，也就剩下图形处理软件等寥寥可数的领域尚未被微软彻底占领。

实现了“人人桌面上都有一台PC，PC里运行的是微软的操作系统”的愿景的微软，貌似是再无对手了，在这二十多年间，IBM的OS/2、苹果的Mac OS、开源的Linux，以及中间很多小型操作系统都曾经站起来想挑战微软的统治地位，无奈用户的习惯是强大的，微软的执行力是超强的，除了Linux等开源Unix在服务器领域和黑客奇客环境里占据了一席之地外，没有人能从根本上撼动微软在桌面系统上的霸主地位。

于是微软的操作系统优势也逐渐演变成微软世界第一软件公司、Bill Gates世界第一富豪的强大基石。

操作系统？

这个霸主地位同样也受到无数研究人员的思考：是什么让微软保持了这个霸主地位？如何能动摇它的地位呢？大多数研究人员，都得出类似的结论：如果一个系统能像Windows那样易于获得，并且使用上延续用户的习惯，那么应该是可以撼动这个地位的。所以，不管Linux的桌面也好，还是中小厂商自己封装的某种系统也罢，通通都在这上面做足了工夫。一个个长得比Windows还Windows。

我想起有一次在某个KTV里点歌，点歌系统突然崩溃，露出黑呼呼的控制台界面，我赫然发现，那个点歌机运行着一个 gentoo的 linux。只是因为手头没有键盘，否则我相信我会很开心地在上面四处查看一番。这个例子很深刻地说明，如果应用软件使用上不存在沟壑，用户其实并不是很在乎具体的操作系统是什么。

可是，为何在很多新兴的操作系统学足了Windows的情况下，Windows依然占据着统治地位不被动摇呢？这里的因素不少，撇开商业和运营的因素不谈，用户的惰性也是很重要的一个因素。这个惰性并非是说用户在做选择的时候，不愿意选择新的东西，恰恰相反，记得这两年的评测数据表明，在确保使用性一致的情况下，在新旧系统面前，用户会做出很随机的选择：接近1/2的用户会尝试新系统。这个惰性很大程度是微软给用户做好的一个茧：所有的应用软件，

都已经习惯了微软品牌的，那么更换就是很痛苦的事情了。这个痛苦会随着用户数据的多寡而加剧，因为对于用户来说，数据是自己所有的部分，比如自己的邮件、自己的相片、下载的文章、视频等，如果数据没有异常，用户是不愿意冒风险去做根本性的变化的。

说到这里我又想起一个例子：前一阶段和同事一起在淘宝上买了一个以太网硬盘盒，装上个1.5T的大硬盘，就再没有担心不知道如何存储将要成为电子垃圾的文件之虞了。硬盘很好用，只是觉得通过Windows文件共享传递文件有些慢，尤其是在看高清视频的时候。

于是，好奇心驱使自己稍微研究了一下，发现硬盘盒子无非是一个安装了支持ARM CPU的一个gentoo Linux版本的嵌入机器。于是想办法通过SSH和一些配置，搞定了里面的Web Server（一个Lighttpd）和FTP，然后再用MPlayer放视频则流畅无比了（哪怕是通过802.11B的慢速无线网）。但是，同事依然喜欢用Windows 共享，因为已经达到他的目的，并且数据在那，没有丢失。而实际上这个硬盘盒子是跑了一个Linux+SMDB（实现Widnows网络共享）的Linux机器而已，但对于最终用户来说，他们更多关心的是：能否实现功能，如果能，则最好不要改变。

所以，时至今日，Windows 7出现，微软依然是操作系统市场的老大。只是在很多地方，人们并不知道自己面对的不是 Windows，而是其他的陌生的东西。

互联网的崛起

那么这个格局就不会改变了吗？笔者认为，也不尽然。突破某个习惯的最可靠方法，是另外一个更强大的习惯。在操作系统领域，这个更强大的习惯就是联网。

我相信，实现了自己企业愿景的微软没有想到，20世纪90年代后期闹得沸沸扬扬的浏览器之战，虽然是以微软最终胜利告终，但也是给微软的操作系统的末日打开了第一扇门。

互联网经过这些年的发展，已经基本上覆盖了所有计算机用户，网络游戏、在线软件的出现，已经让网民越来越少地依赖操作系统和本机软件，而对网络的依赖越来越严重，搜索引擎对人类信息的整理，以及网络的社区化更是让网民几乎无法离开互联网。互联网的强大在于：

它是一个人类社会的缩影，如果说人类社会的本质是基于沟通交流的种群，那么互联网几乎就是用电缆将人与人连接起来，并且赋予每个人近乎完全公平的信息的工具，在这个平台工具下面，每个人的自我和本我都能得到前所未有的张扬。信息的交流是人类社会的基本需求之一，而互联网则极大地满足了这个需求。近年出现的很多新名词，比如“网瘾”、“网病”就足以窥见网民上网习惯之强大的一斑。我相信，对现在的计算机用户，让他们一天不用Windows，估计没啥的，但是让他们一天不能上网，估计很多人就要命了。

因为承载着信息交流和沟通的互联网，显然要比承载着信息处理和存储的操作系统更为吸引人——虽然后者近乎是前者的基础。这个问题，随着最近的云计算的再次流行，又有了更强的注解：操作系统已经离用户远去了。

现在的计算机，尤其是桌面的计算机，很可能没有安装Windows（比如大多数手机和笔者前面说的硬盘盒），但是基本上都安装了某个品牌的浏览器。这个定义，基本适用于智能手机等移动设备环境中。在桌面环境里只是近两年的事情。

作为一个互联网行业的从业人员，体会可能比普通用户要稍微早一些，看着自己的笔记本：邮件系统，基本可以用在线邮件，其编辑功能不弱于 Office套件，游戏基本都是网络游戏了，很少玩单机游戏，个人财物，自然是网络银行的天下了，甚至连 PhotoShop，都出了网络版，处理照片看上去足够了，视频音频，基本都直接在浏览器里看了，朋友们都上开心了，聊天都直接web IM了。作为一个最终用户，如果让我想有什么确实离不开Windows的地方，除了要给老板写PPT之外，一时半会儿还真想不出来。所以当编辑老大找来，想写个Windows 7相关的东西的时候，我脑袋里迸出来的第一句话竟然是：XP 不是挺好的么？要7干什么？（在下在Linux 里用虚拟机跑了个 XP，以备不时之需）。

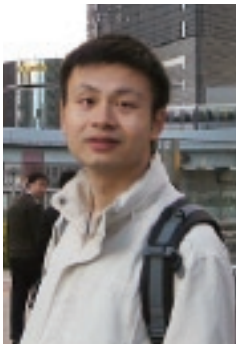
做为一个互联网从业人员，可能领先一般大众消费者那么两三年，所以，也许我说的是大众消费者三年后的感觉，我也相信，在很长一段时间里，Windows 还会是最主流的操作系统。但是有一点我坚信：更为强大的上网、联网的习惯，必将取代用户之前的其他老旧习惯。

小结

五到十年后，操作系统仍然是非常重要的计算机部件，因为浏览器和很多网络应用需要操作系统的支持，但是用户将会渐渐忘却它，对它的选择之路将越来越宽泛。所以，也许离“Windows 7诚可贵，IE价更高，若为上网故，二者皆可抛”的日子真的不远了。

Vista 栽树，Windows 7 乘凉

■ 文 / 杨钢



作者简介：

杨钢，2001年加入金山软件至今，参加了自WPS 2002以来历届版本的开发。现任WPS研发部技术总监兼首席架构师，负责开发部管理和技术框架工作。致力于将架构设计变成一个有理论基础，可复制的过程。

不论你是否喜欢，今年在各IT媒体上露脸最多，受关注度最高的软件产品，就是微软即将（写稿时）上市的新一代操作系统——Windows 7。鉴于Vista不太成功的市场表现，微软对其继任者寄予厚望。而根据微软产品的一个规律，即每一个重大技术革新版本后的下一个改进版本都会有非常成功的表现（如 Windows XP），也难怪大家反应那么积极，对其充满期待的人大有所在了。在这里，笔者也凑一回热闹，从技术和自己的试用感受上，分析一下新一代操作系统对于办公类用户带来的变化和影响。

Windows 7使用感受

从Vista到Windows 7，首先给人最直观的印象是速度的提升。笔者没有做过详细的评测，仅从主观感受来看，后者在内存占用和常规操作的响应上，都有可以察觉到的改进。不过需要说明的是几个月前发布的Vista SP2在效率上又有了进一步的提升，从一些媒体评测来看，与Windows 7相比差距非常小。但是考虑到 Windows 7 SP1可能也要不了多久就会发布，在一个新的基础上提高的余地从理论上说会更大。

另一个不得不说的特性就是著名的UAC了。虽然从技术上来说这个特性可以显著提高系统的安全性，但其弹出的频率之高使得大部分Vista用户不堪其扰。而Windows 7在这方面也做了非常大的改善，不仅对弹出时机作了选择，还提供不同的安全级别备选。从笔者半年的实际使用体验来看，并没有受扰的感觉。

从界面改进来看，Windows 7相对于Vista除了任务栏变化较大外，其他主要是细节的调整。从各方面的反馈来看，这些调整被接受的程度还算比较高。另外，Windows 7还新增了一些功能，网上有很多资料，这里不再赘述。

至于应用程序兼容性，由于有Vista的铺垫（Windows 7的内核版本是 6.1，相对Vista只是一个小版本升级），绝大部分使用广泛的软件已经不存在兼容问题了。从使用感受来看，Windows 7准备得比较充分，用起来也非常顺手。

Windows 7认证之路

作为办公软件开发者，笔者更为关心的是Windows 7的出现，给办公类用户带来的变化。因此，从MSDN发布Beta版开始，笔者一直在使用Windows 7。下面结合实际的使用情况作一些分析。

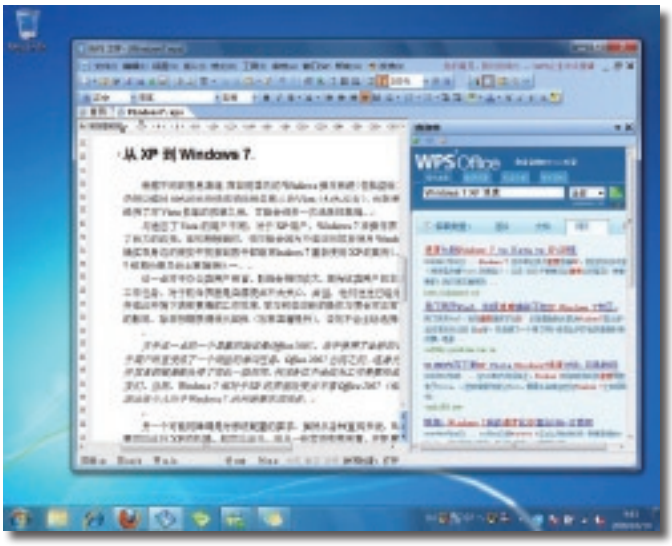
从架构上来看，由于Windows 7沿用了Vista的内核，因此对于普通的应用程序（不需要安装驱动程序）而言，如果在Vista下可以正常运行，则迁移到Windows 7需要做的工作很少。以WPS Office为例，Vista上市前，我们前后经历了几个月的时间才通过Vista兼容性认证，应用程序经过了较大的修改。而这次我们仅用了两周时间，在 7月初就完成了Windows 7的兼容性认证。在开发工作方面，我们仅仅对应用程序界面进行了微小的调整，内核没有变动。由此来看，从 Vista到Windows 7，的确是一个前人栽树，后人乘凉的工作。

在Vista认证过程中，不得不提的是UAC。由于操作系统安全架构的根本性改变，应用程序不得不对其部署模型进行非常大的调整，以符合新的平台和技术。对于我们，这意味着，原本被集中在一个应用程序中的逻辑和功能，因为操作对象和操作本身的性质不同，需要分离开来以便系统区别性对待（UAC的判断是针对可执行文件的）。这不仅意味着需要增加可执行文件的数目，更关键的是，原本属于一个整体流程的各个操作序列，可能需要分布到若干个不同的可执行文件（也就是系统进程）中。这样一来，原本简单的函数调用即可完成的工作，需要使用跨进程通信来替代，增加了应用程序的复杂度和研发工作量。

除此之外，Vista兼容性认证对于各种文件的存放位置也做了相应的规定。这样做的目的是规范应用程序对于系统文件夹的使用，只在系统文件夹中放置必需的内容。因为WPS Office之前已经是这样做的，因此没做什么修改。

另外，除了兼容性，Vista认证对应用程序质量也有较高的要求。在认证过程中，微软提供了很多系统级测试与调试工具，以检查应用程序中是否存在潜在的影响系统正常运行的因素和问题。这一点虽然与兼

容性无直接关系，但同时对应用程序质量的提高起了一定的帮助作用。



WPS在Windows 7下的运行效果

现在回想起来，虽然在兼容性改造上，我们付出了非常大的努力，但是通过配合操作系统进行应用程序的调整，有助于帮助操作系统正确识别应用程序行为。从单个应用程序来看可能没有什么变化，但是当大部分应用程序都完成了这个升级改造工作之后，确实可以帮助操作系统提升整个应用环境的安全性，使广大最终用户受益。

从XP到Windows 7

根据不同的信息渠道，在目前国内的Windows操作系统（包括盗版）用户中，Windows XP仍然以超过90%的比例远远领先排名第二的Vista（5%-6%）。也就是说，绝大部分用户在经历了对Vista数年的观望之后，可能会迎来一次选择的高峰。

与适应了Vista的用户不同，对于XP用户，Windows 7在操作界面和使用习惯上进行了巨大的改变。在初期接触时，很可能会因为不适应而放弃使用Windows 7（事实上，我也确实在身边的朋友中观察到若干卸载Windows 7重新使用XP的案例）。这可能成为Windows 7短期内普及的主要障碍之一。

这一点对于办公类用户而言，影响会相对较大。因为这类用户的主要目的是完成自己的工作任务，对于软件界面是否漂亮并不太关心。并且，他们往往已经适应了一种操作习惯，并在此环境下具有更高的工作效率，学习和适应新的操作习惯会对正常的工作效率产生负面的影响。除非预期获得很大回报（效率显著提升），否则不会主动选择改变。

关于这一点的一个显著的验证是Office 2007，由于使用了全新的Ribbon界面方案，对于用户而言变成了一个艰巨的学习任务。Office 2007出现之初，连身为办公软件WPS Office开发者的笔者都头疼了很长一段时间，何况各位不会因为工作需要而去研究Office软件的朋友们。当然，Windows 7相对于XP的界面改变并不像Office 2007（相对于2003）那么大，因此我个人对于Windows 7的判断要乐观很多。

另一个可能的障碍是对系统配置的要求。虽然从各种宣传来说，Windows 7的目标是只要可以运行XP的机器，都可以运行。但从一些实测数据来看，在配置较低（特别是内存较少）的机器上，Windows 7的响应速度还是不如XP。并且，能够安装并运行Windows 7，和可以流畅的运行各种任务相比，还是有一段距离的。不过，根据个人经验，只要适当增加内存，情况可以大为改观，因此这是一个容易解决的问题。

当然，以上只是列出了可能存在的一些问题。从技术上而言，Windows 7不论是相对于XP，还是Vista，都有很明显的改进。这一点在用户选择时，也是一个非常重要的元素。

综合评价

在本文最后，笔者想就是否升级到Windows 7的问题上，谈一下个人见解。

对于喜欢追新的朋友而言，Windows 7是值得期待的新一代操作系统，不论是界面还是功能上的改进都有很多可圈可点之处。

对于需求明确，以满足自己要求为目标的朋友而言。如果正在使用Vista，那么建议升级到SP2而不是Windows 7（免费除外），因为相对于Vista的改动并没有那么大。如果正在使用XP，则建议安装试用并评估一段时间。当然，如果考虑升级硬件，那么购置一台预装了Windows 7（家庭高级版或以上）的品牌电脑是一个不错的选择。

除此之外，如果你是自由软件支持者，请坚持自己的选择，你们也是软件世界中不可少的组成。

总之，Windows 7是一款非常优秀的操作系统。而且，从前半年用户对测试版的反应来看，它确实赢得了很多正面评价和用户的期待，已经有了一个非常好的开端。当然，市场是最终的决定者。过几天Windows 7就要正式上市了，让我们拭目以待它的精彩表现吧。

在 Windows 7 中寻找渐渐远去的 WinFS

■ 文 / 刘如鸿



作者简介：

刘如鸿，擅长大型系统的架构与互联网产品设计，目前为支付宝上海研发中心架构师，专注于系统架构设计。

2003年9月的微软PDC大会上，微软第一次介绍了代号为“Longhorn”的下一代的操作系统，也就是我们今天所熟知的Windows Vista，在这个号称全新的操作系统中，包括了三个重要的子系统：Avalon、Indigo和 WinFS，分别为图形渲染、消息通信和文件存储。只是可惜在一次又一次的跳票之后，姗姗来迟的 Vista带来了绚丽的图形界面，却没有带给我们过多的惊喜，对于硬件要求的苛刻和不兼容，加上性能方面的不尽如人意，从某些角度说并没有达到当初微软对它厚望，也注定了无法成为像Windows XP那样对于业界产生深远影响的操作系统。

2009年10月22号，微软在全球正式发售Windows 7，也就是业界大多数人认为的Windows Vista升级版，相比Vista上市以来的恶评如潮，Windows 7从发布以来就赞美多于批评，再去介绍哪些新功能，再去分析Windows 7将会对于业界产生怎样的影响，一个我害怕人云亦云，另外一个则是不负责任的。不可否认，Windows 7对于Windows Vista带来的用户体验和性能的提升，但如果现在就去下结论这是一款跨时代意义的操作系统，有些为时尚早。

正因为如此，我们不如回过头去看，因为历史是真实的。一眨眼时间已经过去六年，当年三个令人激动的子系统也已经面目全非。在2006年，或许是为了抚平抱怨不断的开发者社区，更或许是为了不断跳票的“Longhorn”挽回一点颜面，微软宣布将提前发布这些组件，而不是和新版本的Windows一同发布，微软将其中的Avalon和Indigo再加上其他的一些功能整合发布为 .NET Framework 3.0，作为 .NET 2.0的一个补充，而这个版本的 .NET Framework实际上是WPF、WCF和WWF的整合，而 WPF和 WCF的前身正是我们期待已久的 Avalon 和 Indigo。从某种程度来说，微软也算兑现了部分诺言。

可是WinFS去哪儿了呢，微软在2006年6月宣布在Vista发布中将不包括WinFS，但是一些关键的技术和组件将会集成近ADO.NET和SQL Server，正当所有人认为 WinFS已经是微软又一个胎死腹中的项目时，同年11月CEO斯蒂夫·鲍尔默宣布WinFS “正继续开发中”，但没有一个明确的发布时间。

刚刚发布的 Windows 7在许多方面都有显著提升，略感失落的是依旧没有WinFS，可究竟什么是WinFS的，这个被人渐渐遗忘的代号又有怎样的前世今生，这么多年以来又对微软的技术产生了怎样的影响，那么就同我一起趟过时间的河流，去寻找渐渐远去的传说。

WinFS的全称为Windows Future Storage，也称为Windows未来存储系统，在2003年微软开发者大会作为Longhorn的子系统宣布，WinFS是其开发代号，被设计来持久化和管理结构化、语意结构化及非结构化数据，它支持层次、关系和面向对象三种存储关系，当初微软对它寄予厚望，希望其是继NTFS之后的全新文件系统。

说到WinFS，我们还得回到1991年的微软PDC大会，时任副总裁的Jim Allchin（前微软三号人物，Windows之父）宣布了一个代号为Cairo的开发项目，也就是Windows NT 4.0的开发代号，在这个项目中，OFS(Object File Sysem，对象文

件系统)作为关键的一部分，在1996年这个项目搁浅，OFS也就随之夭折，这是我们能够知道的关于WinFS最早的故事了，可以看到微软在很早之前就开始考虑建立基于对象的文件系统。

然而在后来的COM开发过程中，一个称之为Storage+的文件系统被提上了计划，意在为所谓的SQL Server 8.0提供存储基础，后来的SQL Server 8.0不了了之，在开发 SQL Server 2000的过程中，又搞出了一个非常相似的技术，名为RFS(Relation File System，关系型文件系统)，最终上线的SQL Server 2000只是7.0的一个小升级罢了，RFS自然销声匿迹了。

事情并没有结束，假若如此，也失去了我继续说下去的意义了，在2003年的PDC大会发布的版本号为 Build 4501的 Longhorn就包含了WinFS，这也是微软第一次将这个名字推出来，因为碰到了一些无法克服的性能问题，在 2004年 8月微软宣布将不会在新版本的 Windows中集成，而是作为独立的下载包提供。

2005年9月29日，微软悄悄地发布了WinFS Beta1，并允许MSDN订阅用户下载，它可以运行于Windows XP，并且需要 .NET运行时，在同年12月1日又发布了针对 .NET 2.0的Beta1版本，微软许诺在2006年的某个时间将会发布Beta2，正当所有人以为WinFS将会集成进Desktop Search(桌面搜索)时，然后 2006年 6月 23日，微软意料地宣布 WinFS将不再作为独立的产品发布，其相关的技术会集成到其他技术当中。这些技术后来对于微软开发者社区产生了深远的影响，如ORM演进程现在的ADO.NET Entity Framework，而非结构化的文件系统对象访问则被集成进了SQL Server 2008，我们可以通过 FILESTREAM的数据类型得到这方面的支持，而同步组件则被用于现在的Microsoft Sync Framework，当然更多的技术则消失了。

几乎所有的分析师和评论家认为WinFS应该寿终正寝了，可11月鲍尔默宣布依旧在开发中，又让许多人对于WinFS的未来燃起了希望，有分析评论认为WinFS更加适合作为技术孵化器，而不是单独的技术项目。到目前为止，微软对于WinFS的开发没有明确的表态，从Windows 7的推出没有关于太多文件存储的亮点，我们可以认为2006年之后WinFS的研发团队接近解散，或者作为SQL Server 研发团队的一部分，WinFS终成了海市蜃楼。

2008年5月MSDN Channel 9发布了一篇名为《Where is WinFS now?》的采访，采访对象是原 WinFS的开发主管 Quentin Clark，也是现在的SQL SERER业务单元的总经理。我们了解到 80%的开发人员留在SQL Server团队中，而WinFS所孵化出来的许多技术被逐渐地集成到SQL Server当中，而当初 WinFS的许多设计理念也对于SQL Server的设计产生了重要的影响。

说到这里，WinFS似乎可以告一个段落，虽然到目前为止并没有真正取得成功，但是在Windows 7的新功能上也看到了一些 WinFS的影子，也就是Library(库)。对于大多数人而言，界面的华丽的确可以吸引眼球，但落到实处还是文件处理，我想大多数人都有过找不到硬盘上的文件而苦恼，虽然Google 桌面搜索和微软的桌面搜索能够帮上许多忙，但总感觉还是不足，虽然从设计理念上来说WinFS正是来解决文件管理的问题，也许理念过于超前，也许真存在不可克服的技术问题，这种结构化、半结构化和非结构化的数据管理方式也不是一朝一夕能够实现，但依旧值得我们期待。

虽然不能够在文件系统级别上完美支持，但是我们也应该看到正在一点一点地接近，比尔·盖茨在一次采访中提到过微软有计划逐步在Windows Media Player，Windows Photo Gallery，Microsoft Office Outlook等多个应用程序中后端采用WinFS技术，具体时间，依旧是一个未知数。

六年的时间，我们看到当初的Avalon成就了今日Windows7华丽的界面，Indigo也真正成了微软分布式通信框架的基础，唯有WinFS，一路走来步履阑珊，甚至渐行渐远，可回头看看走过的路程，我们有理由相信在未来的某天，这种跨时代的变革终将到来。在 Windows 7上市之际，我们用另外一种方式去看待这个全新的操作系统，WinFS的远去只是这个庞大操作系统背后的一段故事罢了。又不知道有多少充满想象力的技术在天才的工程师灵光一闪中诞生，最终夭折。

当时明月在，曾照彩云归。

价值驱动的组织转型

编者按：随着众多国家和地区众多最新经济统计数据的出笼，世界上越来越多的财经界人士们认为：由金融风暴引发的经济衰退正在慢慢消逝，各种各样的公司和企业也在逐渐恢复元气。然而幸存下来的组织们不能不想这样一个问题：我们应该如何改变自己，才能增强自身的生存和发展能力？我们应该如何改变自己，才能增加自己对于客户的价值？我们应该如何改变自己，才能在类似情况下生存？从本期开始，我们将连载《价值驱动的企业转型》系列文章，为广大软件企业提供相关参考。

■ 文 / 王晓明

Simon 是一家立足于英国本地物流市场的IT公司的技术总监。自从2006年Simon开始接触到有关敏捷方法的相关实践，方法背后的理念及其一些成功的例子，他开始将敏捷带进自己负责的三大产品和多个项目团队，并且进行了多个敏捷实践的尝试，包括将需求以Story形式表述从而替代繁复的文档；与客户每两周一次的需求分析，迭代计划会议；大量使用自动化测试替代原有的手工测试；进行定期敏捷回顾来提高产品和项目有效性。这些积极的尝试正在潜移默化地对一个组织进行着转型。作为一个拥有超过10年的技术经验、成功的资深技术人员和架构师，Simon坚信敏捷辅导是他职业发展的下一个目标。Simon的理想是将这个公司变成一个人人都喜欢工作的地方。

亏损、争吵和瓶颈

Stephen是这家公司的管理总监和拥有人，他对Simon非常信任和支持，由于Stephen曾经做过10年的程序员，他对敏捷的理念也非常认同，并且认为这是公司发展的必然选择和竞争优势。这个对于公司成立16年以来最大的转型项目从2006年开始悄然进行，或许没人真正知道这条路会走到哪里，但是令大家最担心的问题还是发生了：公司最大的定制产品项目开始亏损，客户开始流失，导致公司的财务状况出现了问题。

这是一个令人困惑的现象。敏捷的实施理论上应该带来更好的客户满意度，在同样成本基础上带来更大的商业价值。为什么在这个公司发生的却是恰恰相反的情况呢？与此同时，客户开始抱怨“敏捷”不够灵活，没法时刻解决其最紧急的问题，敏捷的花费也比传统方法大很多。公司的内部开始对敏捷产生质疑，团队开始产生了不信任。Stephen和Simon之间也产生了隔阂和争吵。

在2009年初，Stephen邀请我来做这家公司的运营总监，并且负责公司的敏捷转型。当我对这个最大的、却亏损的定制产品项目进行分析和调研后发现，团队内部已经开始“敏捷”了，而客户方依然在PrinceII的框架下管理产品和项目，极其“不敏捷”。让我们通过图表来看一下其管理的全流程，见图1。

这种项目管理方法的假设前提是：任何业务团队的需求都不能有变化，软件开发团队的效率始终保持稳定。众所周知，对于这种业务模式和客户不断变化的公司里，对IT的需求量也是不断变化的，因此这两种假设都完全不成立。由于客户方依然固守这种在错误假设的基础上建立的项目管理和采购流程，导致了客户和软件开发团队的争吵、客户内部IT团队和业务团队的不断争吵和不信任。业务部门开始压低采购软件产品的价格，从而直接导致软件产品提供商的销售额大大下降。软件开发团队无法有效地将变更的高优先级需求迅速实现成产品，从而导致客户业务部分失去了市场的先机，从而进一步压低成本。这种恶性循环导致了“双败”。

这是一个组织转型，尤其是敏捷转型中的常见例子。IT部分潜移默化地改编自身的工作方法、交流沟通方法，与此同时却没有与客户或者业务部门达成共同的理解和认识，客户无法看到这种改变会给其带来的商业价

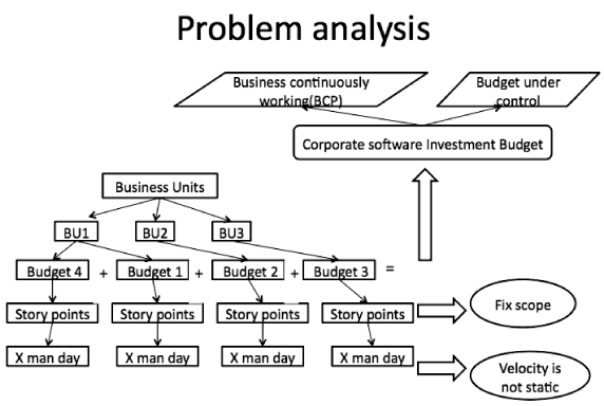


图1 带来亏损的定制产品项目管理的全流程



图2 项目管理和采购流程中的恶性循环

值，因此很难接受这种工作方式的转变。由于转型中的一方过于关注流程、技术实践、项目管理形式，而忽视了转型的驱动核心“价值”，从而导致了这次转型遭遇重大的挫折。这个具体问题的解决方法并不复杂，就是将不必要的层级去掉，将来自各个业务团队所有的需求放到一个唯一的“需求流”，对需求以Story的形式进行表述并标明其商业价值，进而确定每个Story的优先级，将高优先级的Story计划到最近的迭代中。

这种解决方案是“价值驱动”的，并且有效解决了各方头疼的问题。客户方IT总监不需要再担心预算超支或者ROI的价值，因为从整个公司角度，无论什么时候都是可以创造最大商业价值的需求最先被实现。这种解决方案也会对整个产品团队各方的角色和职责重新定位。

业务部门：

职责：持续地给Product Owner提供高优先级的需求并和软件开发团队共同分析和确定需求。

好处：不需要担心预算的超支。

Product Owner：

职责：管理和维护一个唯一的“需求流”，和业务部门共同确定需求的优先级，计划迭代并且帮助软件开发团队提高效率。

好处：不需要担心业务部门的抱怨。

软件开发团队：

职责：与Product Owner共同计划迭代和产品演示；帮助客户方找到，分析需求，问题；设计，实现和测试软件产品；这个过程中保证服务的质量。

这样的转型给产品团队的各方都带来了好处，消除了浪费和不必要的误解、冲突。



图4 产品团队各方达到均衡

这个方案的成功实施，大大改善了公司与客户的关系，为客户创造了更大的商业价值，并且更早地交付了高优先级的产品。客户方面看到IT投资的回报提高了，更加增强了与这个公司合作的信心和信任。与此同时，公司由于需求的稳定进入，帮助客户创造了更多的利润，其销售额和现金流都有了非常大的改善。自己最大的产品项目也实现扭亏为盈。最终达到了一个“双赢”的结果。

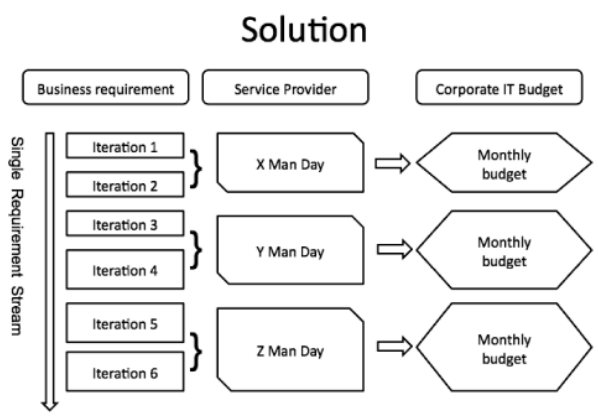


图3 敏捷转型危机的解决方案

组织转型的衡量标准

回头我们来看一下IT公司在敏捷组织转型中遇到重大挫折的原因。

首先，在转型过程中，该公司没有按照“价值驱动”的基本思想进行分析和决策，反而使用了流程驱动，忽视了产品部分和客户方业务、市场、销售的有效结合，从而导致了多方的误解和大量浪费的产生。

那如何解决这种转型中出现的类似问题呢？

首先是要用“价值驱动”的理念来指导转型项目；其次对转型项目中的各个任务进行ROI的分析，来确定是不是可以创造价值的任务。转型的各个阶段可以使用价值流分析方法来不断优化所改变的组织结构、流程和工作方式等。

盈利是任何公司的首要目标，一个组织转型的目标也应该是实现盈利增长或者盈利能力的增长。这就有一个如何衡量的问题。盈利的增长比较容易衡量，可以通过单位时间内的平均利润来衡量。盈利能力的衡量一直是一个难点和热点。

经济学家Ronald J. Baker给出了如下的等式：

盈利能力＝智力资产＊价格＊有效性

其中，智力资产还有以下几本分组成

智力资产＝关系资产＋人力资产＋组织资产

从以上的公式不难看出，为了提高盈利和盈利能力的转型项目是要具体提高“有效性”，“关系资产”，“人力资产”和“组织资产”，尤其是最后一项可以从一个转型项目得到本质的提高。

由于在这个连载后面的几篇文章中我会对组织转型，尤其是敏捷组织转型有更为详尽的分析和解决方案，在这里我就只做简单解释了。

这种以提高盈利和盈利能力为目标的组织转型需要自上而下的规划、指导、方向把握、目标管理，并且需要自下而上的实施和推动。这样才能保证转型过程中的目标一致，不产生脱节现象。

敏捷组织转型有很多种方法，当然核心问题是怎样更好实现价值驱动。因而就需要评估一个组织各个层级和



图5 价值流分析方法

元素当前的形态，继而根据组织盈利能力和成熟度标准确定目标，制订计划，实施，最终进行有效的检验和衡量。

对于评估的结果可以进行分阶段，分步骤的计划，并根据进展不断地完善计划。

认清敏捷组织转型的风险

组织转型中组织结构、工作流程、人员职责、角色固然重要，对人的改变、对各个方面的沟通协作方式的改变、如何更好的激发组织成员的创造性、专注能力和承担责任的态度才是更为重要的。

想使用敏捷的理念、方法为组织创造更大价值的组织转型，就要首先清楚地知道一个什么样的组织才是敏捷组织。我个人对敏捷组织的定义如下：

敏捷组织是一个强盈利能力、有效的、精益的、互相信任的、自我激励的和学习型的组织。

一个敏捷组织的转型需要增强其在11个横向维度和一个纵向维度的成熟度，从而达到对整个组织盈利和盈利能力的提升。

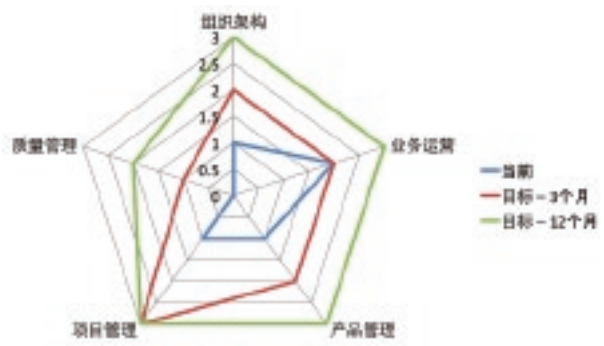


图6 盈利能力的衡量

企业结构转型是一个高风险的项目，即使是一个小企业也是如此。尤其是当你不知道如何进行企业转型，企业需要达到何种敏捷成熟度的时候，这个风险往往会更大。

那为什么说组织转型是一个高风险的项目，转型与一般的变更有什么不同呢？

与“变更”是情形的（situational）不同，“转型”是一种更为复杂的、和逻辑学与心理学密切关联的、高风险的组织行为。当转型被简单地作为一种变更来对待的时候，风险会更高、更难预期。为了让转型项目低风险并且可控，敏捷组织转型需要整个组织在共同的策略、规范、框架下进行。

因而一个转型项目应该遵循以下的策略和原则：

转型的过程是价值驱动的。在整个转型过程中，每一个变更都需要分析，确定这个变更直接产生商业价值或者提高盈利能力（有效性和知识资产）；一个转型项目需要高级管理层的监督指导，并且需要自下而上地开展。高级管理层负责设定目标、提供指导和组织资源；自我完善的渴望和转型的驱动力往往来自底层。

当通过敏捷成熟度的评估，知道该组织当前状态（where you are）并且可以确定该组织的敏捷成熟度目标（where you are going）的时候，就需要使用一些工具和方法，包括：辅导、培训、Workshops、定制规范等。同时还可以利用其他的资源检查表和模板等进行辅助。

不同的方法和工具是应用于不同的情形和人群的。

培训：一般适用于普遍性的概念、原则、原理的介绍、经验的分享。需要参加的人数比较多，可以采用独立或者结对的方式。

辅导：一般适用于需要根据实际情况分析、解决的问题和一对一、一对多小范围内的指导和帮助。通常采用结对的方式。

Workshop：一般适用于敏捷项目中所需的一些workshop的演示，共同完成真实的workshop。例如需求采集、分析的workshop、价值流优化的workshop、写story的workshop等。

定制规范：一般适用于根据该组织的实际特点和情况，基于敏捷组织的要求和敏捷的基本原理进行定制的规范制定，或者在已有规范上的优化。同时疏通流程，保证规范在实际中的应用符合要求。

如我上文所阐述的，组织转型是一个复杂的、高风险的项目，针对其中的更多细节我会在这个连载的下一篇文章对此进行更为详尽的分析和阐述。并且会对敏捷组织中的产品管理部分与读者进行深入的探讨。

译者简介



王晓明，一个组织转型方面的专家，拥有作为IT运营总监，产品，项目经理，组织转型咨询师，敏捷教练和商务分析师8年的工作经验。他拥有英国约克大学的硕士学位。在过去的5年当中，王晓明在多个的企业级信息系统开发项目中工作。这些项目涉足金融，物流，零售，机械加工，石油天然气，安全环境管理，电信，CRM, CMS, BPMS和网格计算（云计算）领域。AAladdin.com 是王晓明创始的针对一个组织转型到敏捷组织的转型框架。

责任编辑：郑柯 (zhengke@csdn.net)

又当爹又当妈的产品经理

一个成功的产品就像一个人的成长过程一样，会经历孕育、婴儿、少年、青年、成年、老年等几个不同的阶段，而产品经理就像无数操心的父母一样，在不同阶段要有不同的心态和处理方式，以帮助产品茁壮成长。

■ 文 / Eric Sink　译 / 顾全

在我贴出的一篇博客中，有这么一段：

最后说点完全不相干的事，别错过了6月3号微软技术大会上的即兴演奏会。我们几个SourceGear来的小跟班儿打算上台表演一曲Pinball Wizard。我弹原声吉他，我们的开发经理Jeremy Sheeley弹贝司，而我们的产品经理Paul Roub弹Evil Mastermind Schecter PT。

话说回头，那篇博客的第一个读者评论写道：

3个经理。哇。你们公司肯定还在增长，要么就比较大。

发表评论的仁兄很可能是用“经理”一词来指代“管人的人”。如果是这样，那是没错，我们公司的经理不止3个，但是演奏Pinball Wizard的3个并不都符合这一名号：

- Jeremy Sheeley的确是个经理。他管理着Vault和Fortress开发部门。
- 严格说来，我觉得我是经理。但是认识我的人会说，把我当成经理实在太抬举我了。
- 但是Paul Roub是“产品经理”。在SourceGear（还有我所知道的大多数其他公司）这里，产品经理并不[必然]管理他人。

因此，当我看到那个评论时，我告诉自己应该写篇有关产品经理角色的文章，也就是本文。

产品经理是什么？

简而言之，产品经理是市场营销人员，他们关注于战略和产品特征之类的东西。

人们一想到市场营销，念头就跑到标识、图形设计和广告等事情上去了。这些是市场营销中的沟通传播一面而已，也就是所谓的“营销传播”（marcomm）。

市场营销的其他方面是那些更关注产品本身的东西：

- 市场定位
- 产品差异
- 功能特性
- 竞争状况
- 市场研究

这些活动都是产品经理的领域。

我仍然记得自己当初学到这些区别的时刻。那时候我在Spyglass工作，一位同事问我：“我们为什么没有产品营销的人员？”

我说：“你是什么意思？我们有营销人员啊。Marc和他那组人刚花了六个月来确定我们的徽标上该用哪种标准色卡红。这就是市场营销，对吗？”

我一窍不通的说法之后引起的探讨，让人很有启发。

管产品还是管人

“经理”一词令人迷惑。像刚才提到的，产品经理不一定管人。 为了要凸显产品管理和人员管理之间的区别，我打算先简明扼要地讲解一下软件公司里管理人员的问题。

对软件公司人员管理的简要讲解

别把他们当孩子。

这就完了？

对滴，就是这样。别把他们当孩子。

如果你遵循这一规则及其引申，你就会成为能干的经理，并让你在软件业界的经理中百里挑一。软件从业者都是成年人了，他们不该被当成孩子一般对待。

而另一方面……

软件产品跟你的同事不一样，它们应该特别被像孩子一般地对待。它们叛逆任性，需要有严格的界限和大量的引导。 像孩子一样，产品也需要经历各个阶段。Joel Spolsky说：“做优秀软件，十年磨一剑”。这些十年并不都相同。

软件产品的成长历程有六个不同阶段，而期间的进展过渡，非常类似养育子女的过程。

- 每个阶段都需要不同方式。
- 有一个逐步从“严控”到“放手”的过程。

第一阶段：准备

在养育子女时，第一阶段是受孕怀胎。而在软件业里，这一阶段涵盖了第一次产品发布前的所有时间，其

中心思想是“准备”。这一阶段中，你找到一个产品创意，梦想它长成以后的样子。

产品经理在第一阶段容易犯跟新爸爸一样的错误：他认为既然妈妈（开发团队）正怀着孩子（撰写代码），他就什么都不用干，而所有要做的，就是等着孩子大到能去打酱油了，是这样吗？

准备

作为产品经理，第一阶段可能是你最重要的一步。如果想要产品成功，你有许多准备工作要做。

- 你需要找准产品定位。
- 你需要明确定义产品差异。
- 你也需要弄清楚这两点如何体现在产品的其他方方面面。

这富于挑战，而你必须现在就做，以后就太晚了。

失败案例

有谁还记得苹果Pinpin吗？

当我在大会上讲演时，经常被介绍说：“他在Spyglass领导的团队，开发了后来许可给微软成为IE的浏览器”。这是真的，但是却不是真相的全部。

有大概120家不同公司获得Spyglass的授权，发布了全部或部分基于我们浏览器代码的产品。除了一个产品，其他全部玉石俱焚。所以也可以同样准确地（尽管是很不客气地）介绍我是“本星球上与失败浏览器产品的关联最多的人”。

其中一个产品是苹果Pippin（虽然我迫不及待地要说，我的代码不是Pinpin失败的原因）。

Pinpin是产品营销的灾难式范例。它败就败在有人在第一阶段的活儿极其糟糕。该产品的定位从来就不清晰。

- 它是个游戏机吗？
- 它是个计算机吗？
- 它是个机顶盒吗？

对上述问题的回答都是肯定的。Pinpin努力要适合每个分类，结果却成了四不像。可以预见，Pinpin的主要差异化特点就是在每个产品分类都表现差劲。对，它是个游戏机，但是贵得离谱，而且与市面上100%的流行游戏都不兼容。对，它也是个计算机，但是运行缓慢，马力不足。

第二阶段：照顾

在养育子女时，第二阶段是出生和婴儿期。而在软件业里，这一阶段涵盖了1.0版发布和之后的时间，其中心思想是“照顾”。这一阶段中，你耐过产品出世的最后痛苦。

产品经理在新产品初次发布中扮演着非常重要的角色。生育的痛苦由开发团队承受，但产品经理要对启动负责。信息发布准确是十分重要的。1.0版的发布、正如新婴儿的降生：非常非常多的痛苦，接着是稍纵即逝的纯粹幸福感，然后再有好一阵子无休无眠。

大多数B2B软件产品的1.0版本发布，更像是个开始，而不是结束。

照顾

婴儿出生后，照顾的责任压倒了父母。新生命是个完全的依赖者，自己什么都做不了。要是他想凌晨3点吃东西，那也是不得不照做的事。

很多1.0版产品也有相似的需要。用户需要技术支持，有时候非常紧急，时间也非常不凑巧，也是不得不照做的事。

像母乳喂养的妈妈的丈夫那样，产品经理可能不用对所有这些需提供的照顾承担基本责任，但他可以参与其中，这会让他和产品都从这一选择中获益。 在第二阶段，产品经理和新爸爸们通常也分享另一个相通的任务：反省。我所认识的每个爸爸都曾经怀抱着他的婴儿，思忖自己是否真的准备好了承担这份责任。

对于准备程度，产品经理在第二阶段需要自问：

- 产品定位准确吗？
- 产品差异化足够吗？
- 信息发布清晰吗？

第二阶段不是开始去做第一阶段本该完成工作的时候，却是复查的时机。做出改变，这时还不晚。

失败案例

大约十年前，我们公司上市了名叫SourceSurf的1.0版产品。它是一个基于网络的应用程序，用于浏览源代码库。如果你能穿越时空回到过去，问问SourceSurf它长大以后想成为什么，它会洞彻未来地说：“我想成为像Atlassian FishEye那样的软件”。但是SourceSurf没机会长大成人了，1.0版是它第一个也是最后一个主要发布。我们本该做的，是微调我们的策略并继续进行下去，然而当时我们看了看初步结果，决定就此放弃。SourceSurf 1.0做得不好，但是产品概念还是不错的。我们只是没能作出该产品成功所需的细微方向调整。

第三阶段：倾听

在养育子女时，第三阶段是所谓的“可怕的两岁”（有父母都知道，这一时期实际持续到孩子几乎四岁的时候）。而在软件业里，这一阶段通常涵盖2.0版发布周期。其中心思想是“倾听”。

这是反抗叛逆的时期，产品开始展示自己的意志。

我的大女儿在这一阶段时，有天我们叫她去吃晚饭。她悠哉游哉地爬上桌，瞟了眼我们准备的食物，吃惊地倒抽一口气说：“除了黄油，这儿没我爱吃的！”

这几年里显然孩子们需要被管教，但他们也需要被倾听。这是“童言无忌”的阶段，有些话很可爱，而有些则令人烦恼尴尬。不论如何，这一过程是成长的重要阶段。父母需要提供管教，但又不能挫伤小孩的情绪。这经常代表要去倾听孩子所说的各种奇谈怪论。

倾听

产品经理也需要在第三阶段花时间倾听。客户要求很多东西，需要听到这些反馈并将其整合进产品的下一

次发布。在产品进入主流之前，仍有时间微调策略。

失败案例

第三阶段是产品失败的最普遍时期。有非常多的例子。几乎任何没跨过鸿沟而死在坑底的产品都是失败于第三阶段。我最爱用的两个例子是BeOS和Wingz。

BeOS是90年代后期发布的新操作系统。技术上说，BeOS很酷，但是它掉进了坑里，因为在鸿沟另一头没有找到商机。

Wingz是80年代后期发布的电子表格软件。Wingz具备Excel直到今天尚没有的功能。但是要生存，它需要在一小群用户中流行并从此传播开来。我非常喜欢Wingz，但我认为大部分人都觉得它太古怪。

我经常在想，如果这些产品更愿意倾听最早期用户的反馈，并作出方向调整，那么他们或许会成功。这可能意味着，让这些产品变成他们的父母不曾梦想过的东西。

离题：瞭望鸿沟对面

自从我受雇于一家版本控制软件厂商，现在对我来说最有意思的鸿沟情形是分步式版本控制系统。这一分类下的显著例子包括诸如Git、Mercurial和Bazaar等开源工具。今天，这些产品尚未成为主流。它们在某些社区内名声很响，但是从事软件开发的大批公司仍在使用集中式工具。

DVCS工具是否能跨过鸿沟呢？

第四阶段：对话

在养育子女时，第四阶段是从小学到青春期的时间。而在软件业里，这一阶段通常涵盖3.0版本，也就是产品可以被当作主流的首个发布。其中心思想是“对话”。

在这一阶段，父母经常对他们的孩子能做到的事情表现出惊诧：“小Bobby真棒！七年前我们生了这个孩子，现在我能和他真正交流了！”第四阶段来到了，父母突然发现他们的婴儿俨然成了个小大人。

相似的，软件产品在第四阶段已经达到了大部分人都能使用的程度。它们是主流了。先前的发布只在早期试用者和有耐性的人中间流行，但3.0发布已经足够精致到成为大半个目标市场的有效解决方案。

第四阶段时，产品和孩子对竞争的关注，发展到了不健康的程度。当孩子进了学校，他们开始了解，可以将快乐建立在他人痛苦之上。他们需要花费很多年才能摆脱这一影响，学到精神健康和个人安宁是来自于后天教化的选择：不必过虑于他人，而是照管好自己。

类似的，第四阶段的产品可能不该那么过分关注竞争，如果此时还没有获得很好的差异化，也就不再可能得到了。照管好你自己的客户吧。

对话

对父母和产品经理而言，第四阶段最重要的是对话。

这是孩子需要信息的阶段，也是你要说任何事情他们都还愿意听从的最后阶段：父母需要与孩子讨论未来的风险。告诉他们诸如抽烟、性和吸毒以及成为芝加哥小熊队的粉丝等糟糕决定会带来的烦恼和后果。

类似的，第四阶段中产品经理的主要工作是提供信息。你的产品已经是主流了。人们想要它。但是为了让他们意识到他们的需要，你需要为潜在客户提供大量信息。你需要白皮书。你需要演示视频。你需要产品附加文档。诸如此类。

失败案例

作为第四阶段进展不佳的例子，我会引用我自己公司的旗舰产品，SourceGear Vault。Vault很受欢迎，也是我们公司的成功产品。但是要说到提供一个主流产品所需的各种信息，我们就搞砸了。就在Vault 3.0发布前后，这一产品开始通过“口口相传”获得大量关注。人们会到我们的网站获取产品信息，但是我们能提供的太贫乏了。

值得称道的是，我们意识到了失误，采取措施补救。我们招聘了一个产品经理专门处理此事。但我们仍然起步晚了，所以就算到了现在，Vault仍在追赶进度。

第五阶段：平衡

在养育子女时，第五阶段是青少年期，从青春期到高中的时间。而在软件业里，这一阶段通常涵盖3.0版之后的发布。其中心思想是“平衡”。

在这一阶段，孩子不再想让你们叫他孩子了。这是通向成年的转变时期，充满了反叛。没有什么阶段比第五阶段更难熬了。每个父母都希望孩子最终独立，但很少能意识到这一历程将如何痛苦。青少年期与可怕的两岁期非常相似。

当孩子不能为所欲为时，他的愤怒可能十分暴烈。但是，他不是满地打滚、又踢又叫，而是大喊“我讨厌你！”，再摔上卧室的门。

平衡

但正像可怕的两岁期那样，这一阶段也是成长的关键时期。父母必须要达到两全其美的平衡。这时开始要少一点操纵，让少年对自己的生活多一些控制。

第五阶段里，孩子和产品把大量时间花在讨要他们所不需要的东西上：

- 可是爸爸，地球上哪个十几岁的孩子没有无限手机短信套餐呀！
- 可是妈妈，我怎么能再穿同一条裙子啊！我得买条新的！
- 可是爸爸，所有其他字处理软件都有语法检查器了！

作为第五阶段中的产品经理，你需要给产品一些自由。是时候为客户提供一些你多年来抵制的功能了。即使你觉得那功能不是个好主意，只要

它不会毁掉产品，你就应当严肃考虑去做到这一点。

但是你仍旧需要划出些界限：

- 父母一定要开始让孩子成为他们想变成的样子，但是别让他们把自己的生活给毁了。设定足够界限，以保证他们安全完成成年的过程，避免一失足成千古恨。
- 相同的，软件产品在第五阶段发运一个糟糕的版本，也一样可以毁掉自己。产品经理需要成为质量捍卫者。版本3.0如何棒是毫无意义的，如果4.0缺陷多而不可靠，产品的名声可能就永远没法完全恢复了。

失败案例

低质产品毁掉自己的最佳例子就是微软的Visual SourceSafe。时至今日，几乎所有关于版本管理软件的笑话里，SourceSafe还是点睛之笔。它被广泛鄙视，大家通常都认为它不可靠。微软买下SourceSafe之前，它还是个受到景仰的杰出产品。这一产品没有成为最广泛使用的专利版本控制软件，实在是没有天理。SourceSafe重新定义了软件行业，比任何后来者提供的易用性都要多。但是在微软的养育之下，SourceSafe毁掉了自己。5.0和6.0都是灾难性的。我敢打赌SourceSafe是史上最受用户鄙视的赚钱软件。我们公司从全世界对SourceSafe的不满中赚得盆满钵满。讽刺的是，最新版SourceSafe可能不该遭受这么多贬损。我还没用过SourceSafe 2005，但我了解它其实没那么差。他们修复了大量的问题。但是，这一产品已经成了大家都乐意去讨厌的东西，这就为时已晚了。世界已经不可能再信任SourceSafe。

第六阶段：放手

在养育子女时，第六阶段是成年期。而在软件业里，这一阶段涵盖产品成熟的时段。其中心思想是“放手”。这时你作为父母的工作基本结束了。从此往后，孩子的任何错误都是他们自己的问题，你不得不放手了。

同样的，产品步入成熟期后，产品经理也没有太多可做的了。

对父母和产品经理而言，这都是庆祝的时刻，一个长久而时时遇到困难的历程成功结束了。

放手

父母和产品经理经常会觉得很难放手，但是这么做很重要。你是个产品经理，你的工作是帮助定义和塑造产品的特征。当孩子“长大成人”，你就不要再帮他寻找他要成为什么样子，他已经成形了。

让营销传播、销售和支持部门就此接手一切吧。

经典失败案例

产品经理不愿放手的两个最明显例子是微软Windows和微软Office。

这些产品都完成了，完成都好多年了。不幸的是，微软在很大程度上没能找到下一个产品。所以，由于微软没什么好做的，他们不断地向我们兜售我们不需要的新版本，杜撰各种理由强迫我们购买。

自然有些人会埋怨我所说的，史上营收最高的两大产品怎么成了失败案例。我没有这么说。我要说的是，这两个产品在第一到第五阶段都惊人地成功，但这无法改变事实，他们第六阶段的实施实在糟糕。

结语

产品经理与父母都各自承受着大量各不相同的观点的压力。你那里的书店对每一主题各自矛盾的书籍也是汗牛充栋。

本文包含了我对于养育子女和产品营销的大量观点。有些读者可能同意我，而另一些认为我错了。

问题在于，对于养育子女和产品营销，人人都知道怎么做，却没人真正知道怎么做。你四处阅读，然后尽力做好。而在养育子女时，做到最好应该就够了。

这是产品管理与养育子女完全不同的一点：绝大多数父母都是成功的。

很少有完全失败的父母。我们焦虑失眠，可最后事情大都会解决。如果你爱你的孩子，并做出最佳的努力，他们很可能成长得挺好。

相反的是，大多数产品都失败了。市场营销是个光靠努力和“全力以赴”并不一定足够的领域。

Pinpin的失败不是因为没有产品经理——它失败于产品经理是个傻瓜。

微软没在Windows Vista上对我们快马加鞭地驱赶，因为他们不在乎找到新产品。微软企图找到下个产品，他们非常地努力。他们搭上了最聪明的人迎接这个挑战。他们花费成百上千万的美元来寻找，但是从未找到。

而Google找到了。

原文链接：

http://www.ericsink.com/articles/Product_Parenting.html

作者简介



Eric Sink，SourceGear公司创建人，曾在Spyglass公司担任项目领导，开发了Internet Explorer的原型。所著《Business of Software》一书获Jolt大奖。

译者简介



顾全，eBay中国研发中心数据仓库(IMD)开发经理。历史学科班出身，澳洲计算机硕士，工作于数据仓库行业多年。作为eBay首批CSM和交易集市技术部门在全球的首个CSP，他领导了eBay首个Scrum实现，并正在推动整个组织的敏捷化。

责任编辑：郑柯 (zhengke@csdn.net)

BlackBerry开发经历（下）

——PodTrapper项目的经验教训

本文上篇刊登于《程序员》10月刊。文章介绍了作者开发第一个BlackBerry移动应用——PodTrapper过程中学到的经验教训，内容涵盖开发、定价、零售和市场营销等诸多方面。

■ 文 / Marcus Watkins　译 / 罗小平

市场营销

对营销的逐步认识

最初我仅打算自然销售——让用户通过口碑相传发现并使用我的应用，以此实现用户数的自然增长。事实上，PodTrapper是在BlackBerry上收听podcast的唯一选择，因此我的Google排名是非常好的。我非常满足。毕竟，我更喜欢将时间花在增加新功能上，而不是绞尽脑汁想办法诱惑用户讨论或购买我的产品。但在看到一些评论（它们往往能在接下来一周左右时间内，将销量变为原来的两倍甚至三倍）后，我认识到了营销的巨大作用。

如果要问我在PodTrapper中得到的主要经验，那就是“营销即一切”（以及“一切即是营销”）。我可以创造出不为人知的最好的产品，但如果没人知道，又怎么会有人买呢？当然，这个显而易见，但对我而言，不那么显而易见的，则是自然销售与主动营销效果的巨大差异。举个例子，两周前我发布了2.1版本。这次更新不算大，支持多播放列表是其中最大的新增特性。因此我认为“没人会在博客上提到它”。但假设归假设，我还是发了一些邮件希望人们能知道这个版本。结果有几个博客认为值得关注，这四封花了我10分钟时间的邮件，起到了如图1所示作用（见红色箭头处）。

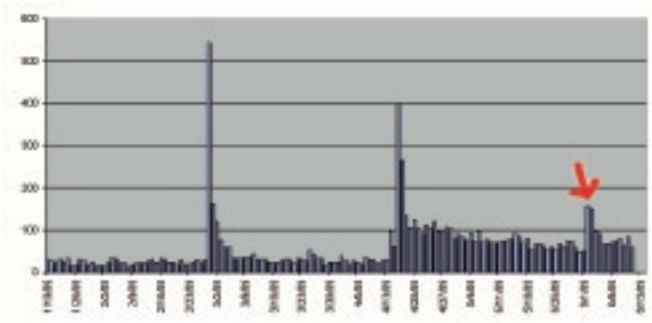


图1 突增的2.1试用用户

如果只按惯例，我悄悄发布新版本，只有现存用户通过程序内部的版本检查功能发现它，就不会有这种效果。这充分说明，只有付出，才会有相应的收获。当然也不是说你有付出，就总能得到回报。再告诉大家另一个窍门：在刚刚举的这个例子之前，我已开始向那些我认为会对PodTrapper产生兴趣、并能帮助扩大知名度的人主动营销。以twit为例，这是一个拥有大量订阅者的podcast站点。这些人花很多时间讨论iPhone上新的podcast下载能力（和PodTrapper提供的功能相比，它自然黯然失色；而且BlackBerry的用户比iPhone更多）。他们当然也会在自己的文章里谈及用在其他平台上的播放器！

> Hi Leo：
>
> 我不知道你是否会看到这封邮件，也许你从来不读你的邮件。我仅想让你知道我已经发布了PodTrapper（BlackBerry上的podcast播放器）的新版本。它能实现你希望手机podcast播放器拥有的任何功能（比如下载、自动管理等等）。
> 如果你有兴趣，请访问我给你的地址：
> http://www.versatilemonkey.com/PodTrapper_the_Blackberry_Podcast_Manager_Gets_a_Facelift.html
> 不管怎么说，你是一个podcast爱好者，而我编写了这个podcast播放器（仅有的一个，现在还真没有其他可选择的同类产品），我想我应该写信告诉你。
> 浪费你的时间了。
>
> Marcus Watkins
> VersatileMonkey.com

我发了很多这样的邮件，每封都专为我要发送的目标用户而写。回复的不多，远少于被我列为目标用户的数量。我仅将那些我认为知道我的应用后，可能会让他们受益的人列为目标用户。后来我偶然看到engadget

fame 的 Ryan Block 写给 PR People 的信。我意识到这些人可能经常收到很多类似“嘿，你可能对我做的这个好东西产生兴趣”这样的垃圾邮件。给大量陌生人随机发送邮件虽然有一定效果（毕竟，我确实得到了几个回复），但效果并不会总有的。要想有效果，必须好好努力，将这件事情做扎实。我有一张电子表格，专门记录何时何地我向谁发了什么，这样做的好处：1、我从来不向同一个人重复发同样的信息；2、我能看到相应的效果。到目前为止，我还没找到传播品牌的更好办法。如果你想到了，请告诉我。

值得一提的是，我至今还是很讨厌批发商（批量发邮件）模式。我花了很多时间，保证我发给每个人的邮件都是专门为他而写，并且此人确实对 PodTrapper 感兴趣。我访问过一些公共关系网站，在那里你可以提交新闻稿，这些网站会给很多人发送邮件。尽管我知道这确实能产生作用（的确有人通过垃圾邮件得到了投资回报的），但这不是我想干的事情。

最后对新闻稿做点说明：还是应该写一份。虽然一些大网站谈到你的应用时，都会自己撰稿，但其主要内容还是从其他地方粘贴复制过来的。而新闻稿，恰是这些内容的一个重要来源。

入驻App World后的销售情况

从市场营销角度来说，App World 对我具有重大意义。4月18日，我被App World接纳，见图2中红色箭头处。

在我被接纳的同一天，约有3或4篇博客文章发表了对2.0版的评论，因此图中的数据受到了影响，但进入App World前和进入后的试用平均量间的差异，还是非常清楚的。如果不看峰值，只看平均数的，那么进入App World前，每天的新增试用期用户是26个，而进入后跳到了76个。当然其中有其他影响因素，如新增特性，新发表的博客文章等等，都会提高用户数，不过这些因素的影响量至多不过30个左右。显然，App World会对人们寻找手机应用的结果产生重大影响。图3是利用App World报表功能统计的下载量。

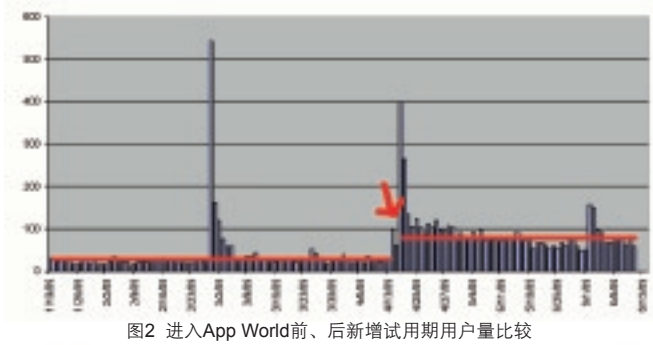


图2 进入App World前、后新增试用期用户量比较

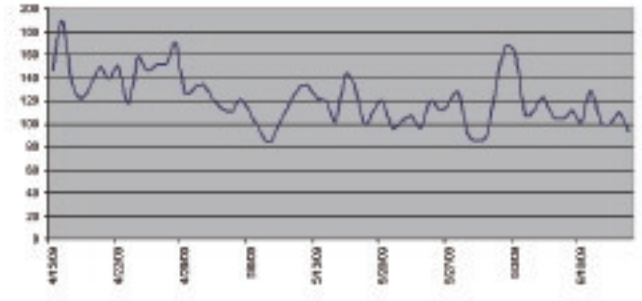


图3 PodTrapper在App World的日下载量

这个图也表明将下载量作为衡量销售情况的指标并不合适。如果你将此图与前面新增试用期用户图比较，会发现二者之间存在很大差异。我认为这主要是不少人重装应用，或更新到新版本造成的。如果仅基于下载量来计算我的注册率，那么因为这其中的差异，我就会低估实际的注册率了。

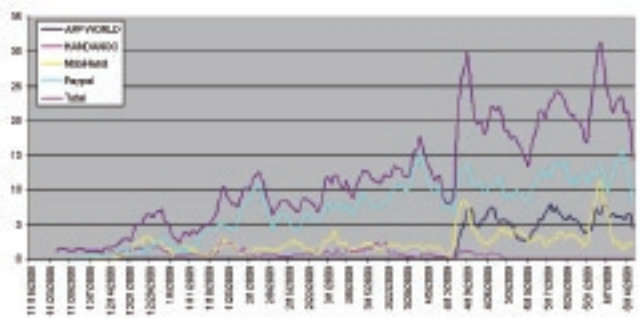


图4 全部销售情况（每天，5日线）

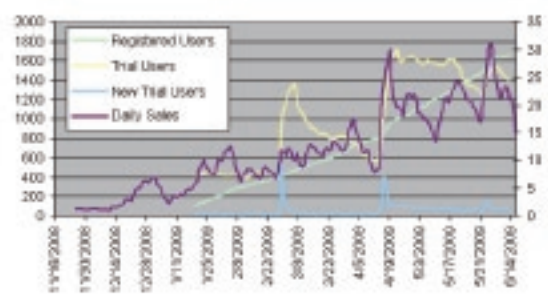


图5 使用与销售情况对比（第二根Y纵轴表示销售量）

到现在为止，我还觉得大多数用户可能都不知道能为他们的BlackBerry手机下载应用。我想一旦RIM解决掉App World目前遇到的一些早期问题后，他们将在这个方面加强营销工作。

接下来是其他一些我觉得比较有趣、和市场及销售相关的统计图（见图4，图5）。我想说的是，如果这种趋势得以保持，那么PodTrapper真的可看做第二份薪水的来源了。第一个图，描述的是迄今为止的全部销售数据，其中的峰值一般出现在博客文章、评论等出现的时候。

我很想知道有关注册的深度信息，相信能在我上面分析过的数据基础上，找到推算出最优试用周期长度的办法，而不是像前面我提到的“30天看起来还不错”。（见图6）

与其他公司（或许不应该）打交道

当我希望和其他一些公司（不是RIM和前面提到过的零售店）合作时，严重受挫。前面提到过，我最后想实

现一个支持广告的PodTrapper版本。我在网络上收集了相关公司的信息，然后给大约10家公司发邮件、打电话或在线填写了申请表，但最后有反应的就有一家，而且是在数周之后。我明白自己只是个小人物，又是个人开发者，用户基数很小，但也不至于根本就没人愿意睬我吧？我连他们不理睬的原因都不知道，自然更不知道如何改正自己的问题。

另一个例子与Audible有关。Audible在BlackBerry平台上提供一个支持long-format格式的播放器。我的不少用户希望PodTrapper本身也能支持long-format格式，可以播放DRM不支持的Audible文件。这是个好主意，那我就开始联系Audible公司吧！但迄今为止我一共联系了四次，没得到只言片语的反馈，哪怕是“那是不可能的，抱歉”呢！

很显然，我犯了个错误。我给人家的印象可能是那种凡事一个人包办的“一人店”。我尝试过各种不同的方法，包括朋友式的“嘿，让我们合作吧”、比较正式的“我们二者的模式在战略上合作，可以获得双赢”，都没有结果。实际上我也是个公司，而不仅仅是个DBA之类个人开发者。但他们没有足够重视，自然也不会发现这一点。我猜测我被拒绝的原因，可能和我在前面提到的那个博客文章作者的情况类似。看来，我的确需要努力提高自己的知名度（可能将现在的名字“Versatile Monkey”改为“VrstleMn.ky”更好，并争取一些风险投资）。

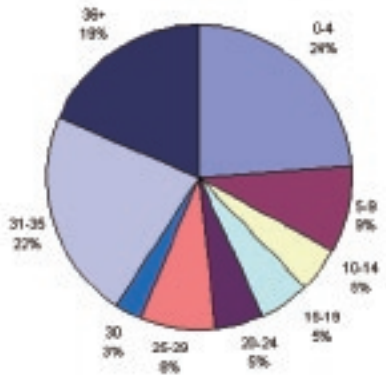


图6 人们会等多长时间后注册（试用期为30天）

我的用户

这是我喜欢的一个部分

在整个过程中，我最喜欢的就是和我的用户之间的交互。如果你读过我博客上的评论或论坛的一些帖子，你就会发现我的用户是多么的棒。如称赞我有如何多的优点，详细论证PodTrapper为什么比同类产品更优秀，以及它真正值得使用者付出$10等。因此，仅仅需要我做，也是必须要做的，是考虑如何取舍用户的这么多想法，并实现到PodTrapper中去。UI的绝大部分细节、甚至一些复杂功能，都直接来源于用户的反馈。

我每天在完成正常上班的工作后，花1小时回复邮件和论坛帖子。我并不是想节省时间，自己的确很想尽可能多回答问题。我收到过很多感谢信，夸奖我回复速度很快。但是，在白天上班时间，我不能做何PodTrapper相关的事情（包括收发邮件）。在这个过程中，我总结出一个窍门，即尽可能先回答最新收到的邮件。是7小时还是6小时后反馈并不重要，但是2分钟而不是1小时后回答就绝对有重大区别。

在与用户交流过程中，我发现不是每个人都善于沟通。

> From: XXXXXXX Fri, May 15, 2009 at 2:32 PM
> To: PodTrapper Support <support@versatilemonkey.com>
> 这个程序就是狗屎，谢谢你骗我，待会再和你说
> Sent on the TELUS Mobility network with BlackBerry
>
> From: Marcus Watkins <marcus@versatilemonkey.com> Fri, May 15, 2009 at 5:16 PM
> To: XXXXXX
> Hi XXXXXXX,
> 很抱歉给你这种感觉，据我所知BlackBerry上还没有PodTrapper的同类应用。我真希望能从你这里听到其他看法。
> -Marcus
>
> From: XXXXXXXX Fri, May 15, 2009 at 5:19 PM
> To: Marcus Watkins <marcus@versatilemonkey.com>
> 抱歉，我没认真了解你开发的程序，它开始已在我的手机上工作了，我喜欢这个程序，你的东西很棒！
> Sent on the TELUS Mobility network with BlackBerry
>
> From: Marcus Watkins <marcus@versatilemonkey.com> Fri, May 15, 2009 at 5:23 PM
> To: XXXXXX
> 好，真诚希望你有任何其他想法时第一时间告诉我让我知道。
> -Marcus

我想自己误解了他第一封邮件的意思——我还以为他抱怨我欺骗了他，待会儿会找律师和我交涉呢，呵呵。收到这样的邮件，我并不介意。我的哲学是没人指出问题当然好，但有时有人指出问题会更好，因为这时候你才有机会展示出你是多么乐意解决他们的问题！其结果可能比产品一开始就完美无缺给用户的印象更为正面。当然也并非总是这样，我绝不会将很容易发现和解决的问题留到产品中，让用户去体验的。一句话，我的意思是我们应该将客户表达的不满和指出的问题，当作能让用户比没问题时更加高兴的机会。即便有时会被一些确实没头脑的邮件激怒，但像下面这样的邮件可以给我带来更多快乐：

> 我必须告诉你，我老婆和我都很喜欢这个应用。它非常独特、用起来很棒。我在我们公司负责管理约4000部设备组成的BlackBerry系统，至今已4年，试过很多应用，都数不过来了。这个应用是我最喜欢的。非常优秀，能最有效利用各种终端拥有的资源。我会向每个人推荐它！谢谢你这伟大的应用；我希望你能赚大钱，参与慈善事业，退休后滨海而居，和Bill Gates、Eva Longoria一起参加法国公开赛！
> 这个应用棒极了！

总结

这篇文章比我预计的长多了。我希望朋友们能从中受益，当然作为交换，自然也会让更多人知道PodTrapper。不过你完全可以认为我这篇文章是没有任何私心的。如果想知道任何我没有提到的内容，请在我的论坛里告知，那里是不用注册的。

■ 责任编辑：郑柯 (zhengke@csdn.net)

架构师接龙

黄冬 vs. 邓毅



提问嘉宾：

黄冬，多年软件开发、系统架构、系统运营的经验。长期关注高可用性、高可扩展性的系统架构设计。主持设计和运行过多个大型高容量产品和系统。是中国 FreeBSD、Python 社区的发起者和积极参与者，也是国内啄木鸟（http://www.woodpecker.org.cn）社区的创始人之一。现在北京从事系统架构咨询及系统运营外包的的创业。

回答嘉宾：

邓毅，网易有道技术总监，负责新技术与应用研究团队的工作，带领很多有道核心技术的开发。清华大学计算机系博士学位，在模式识别、计算机视觉等领域有丰富的研究经验。



黄冬：互联网在过去的数年中发展迅速，交换带宽已经从10M迅速提升到了万M，而计算机的总线、CPU计算能力出现了很多瓶颈。经常会发现一台服务器的带宽有千M甚至更高，而使用的却不足，在架构上有什么考虑，让一台服务器能更多用起这样的带宽来？

邓毅：网络技术的高速发展的确使得单台服务器的带宽有了很大的提高，这可以使得一些以前不太好做的事情更容易的实现。

一方面，对于后台计算服务部分，高速的带宽使得我们可以把更多的服务器通过以太网连接起来，进行大规模的分布式运算，让不同的机器处理大数据的不同部分，再通过网络把数据进行汇总，从而得到以前需要大型机才能获得的超大规模的计算能力，这让我们可以从海量的数据中分析出更加有用的信息。此外，通过网络，我们可以把多台机器联合起来组成分布式的存储系统，从而提高系统的存储容量、访问带宽以及可靠性。

另一方面，在前端与用户直接接触的部分，由于带宽的提升、网速的变快，用户的客户端程序可以更加频繁的与服务器进行通讯，传输更多的数据，“云计算”或者“软件加服务”的模式，使得原来单机服务无法实现的功能或者服务质量得以实现。例如有道的词典，当存在网络连接的时候，客户端的软件会把查询需求发送到服务器端，服务器端可以在一个非常大的词库中进行查询，还可以做一些智能分析的操作，从而在不占用用户本地资源的情况下，大大提升用户的体验。另一个例子是类似“云杀毒”之类的应用，通过网络把数据传回服务器进行杀毒，也是充分利用当今互联网的带宽。

黄冬：作为负载均衡的一种方式，基于四层、七层的负载均衡交换在很长的时间里都是一种非常简单和常用的解决方案，随着服务器计算能力的提升，这种解决方案会发现越来越像是一个瓶颈点，有什么更好的考虑来解决这样的问题？

邓毅：当网站的流量不断变大的时候，负载均衡交换的方法是一个较为简单的扩展系统容量的办法，但是它毕竟是一个通用的解决方案，有一定的局限性。所以，当这个方案遇到瓶颈的时候，就需要对业务流程和用户访问行为进行更细致的分析，找到可能的瓶颈。通常，解决这一些瓶颈的办法来源于对一些冗余运算或 I/O 的优化。例如，负载均衡有可能导致一些重复的运算（可能被均衡到不同的机器上去了），此时通过优化均衡方法、优化系统结构，可能可以在某个地方放入一个合适的缓存，保存中间结果，从而避免重复的运算，最终提高系统的吞吐。再进一步，如果还存在瓶颈，就可能需要在系统架构的层次做妥协，比如通过牺牲一定程度上事务的一致性换取更高的性能。

黄冬：互联网的即时计算越来越多，而这样的计算越来越多的依赖大量的数据存储，比如依据数千万甚至更多的用户使用习惯和用户自身的使用历史，得出该次搜索的排充。在这样的计算面前是否有什么巧妙的方法？

邓毅：使用大量的用户行为数据进行智能的分析，的确能够得到一些非常有用的数据，从而提升用户的体验。当然，这里需要大量的计算资源。

对于一些即时计算的场合，如果想享受这些数据带来的好处，一个办法就是把一些计算模型进行划分：对于规模较大且较长时期不会变化从而可以预处理的部分，在后台花费一些计算资源预先计算好，然后通过分布式的只读系统提供快速的读取服务；中等规模变化较快的数据，可以建立快速计算流程，定时计算，并立即可以访问；而与本次操作直接相关的数据，如用户特征、搜索词等，才是真正的即时计算，这时候需要考虑到足够短的响应时间。

把计算模型分解成这样的三个层次并不总是一件特别容易的时候，有时候甚至需要对计算模型进行必要少量的修改，在不大幅度牺牲准确度的情况下做一些简化，从而可以让不同类型的数据分开计算，最后再综合一起处理。

黄冬：垂直搜索引擎已经成为各个网站及业务系统的必备功能，在这方面是否有一些良好的现成产品和架构方案？

邓毅：随着互联网用户对于搜索这个概念的认识，垂直搜索日益成为各个网站的一个重要功能。当用户不知道某个东西在哪里的时候，可能不愿意一层层的去找，而会去搜。

相比通用搜索引擎，垂直搜索首先需要保证较低的开发成本，因为网站最大的技术力量需要花在其自身的核心价值上；其次，这样的引擎需要满足一定的性能和质量要求，虽然不必与通用过的搜索引擎那样快、那样准；最后，网站通常有自己特有的一些东西，比如一些结构较为特殊的数据需要搜索或者影响搜索排序，这就要求一个结构和功能可以方便定制的引擎。

互联网上有不少优秀的开源项目，我个人比较了解的有 Apache 的分布式存储与计算软件 Hadoop、搜索索引软件 Lucene 等。

黄冬：大量的用户使用行为记录需要留存，而留存下的数据量非常大，对于相关的数据也要进行频繁和复杂的业务计算，对于这样的存储有什么解决方案吗？对于这样的分析型计算有什么有效的架构？

邓毅：首先存储方面，根据计算对于数据的读写需求，可以有两种方案：

一种方式是采用适合流式处理的分布式文件系统，如 HDFS、GFS，通过自动数据块的备份等技术，可以保证数据的可靠存储，同时利用大数据块的存储保证流式顺序数据读写的高性能。当一些数据需要经常性的全量访问，且这样的访问较容易流式的处理的时候，可以把这些数据存在这类文件系统中。

另一种方式是采用分布式的 Key-Value 结构，如 HBase、BigTable 等，他们的特点是可以支持随机读写，而简化的操作限制又使得它比数据库更容易实现大规模的分布式结构，因而适合计算中只需要读写海量数据中随机一部分条目的情况。

运算框架只要是支持大规模分布运算的都可以，例如前面提到的基于 Map/Reduce 的分布式计算框架 Hadoop。

自动化测试结构论

本文基于商业自动化测试工具的研发，分析大量的商业自动化测试工具的实现模式，以提高大家对自动化测试工具本身的认识与理解。

■ 文 / 王东刚

所有的自动化测试工具都由一套非常稳定的、成熟的、持久的测试思想作为支撑，无论QuickTestPro、SilkTest或者TestComplete这些商业测试工具皆是如此。为何本文标题使用了“结构”一词，而非“框架”？我认为，“框架”包容的概念过多，而本文又不打算针对某种自动化测试工具进行整体层面的分析和解构，所以使用了相对描述范围比较小的“结构”一词，用它足矣。

自动化测试的一个经典特征就是对象识别容器，使用对象识别容器内捕获的对象，以实现一系列模拟人工测试的动作，其优点是巨大的，缺陷同样明显存在。对象识别容器内置的识别算法、映射算法、对象排序和出栈算法都让自动化测试工具在执行复杂的、多变的测试案例时，表现得非常耗时和力不从心。另外，对象识别容器因为某些限制，无法知晓它所捕获的对象是临时的还是持久的，所以这直接导致了自动化测试过程难以被跟踪和可重复利用。因此，几乎所有实施了自动化测试的企业都相应付出了极其昂贵的后续维护成本。

回顾国内测试10年，自动化测试应用的大范围推广是必然的，没有一个行业会停滞不前，总有更好更快的技术来提升整个行业的技术水准。在本文中，我会从一个软件自动化测试工具项目（TestMice）展开，描述一些迅捷的、轻量的自动化测试工具结构，例如对象描述、脚本构建、描述性编程、测试数据驱动、场景恢复、插件和编译运行。我不会评价任何测试工具的好与坏，我只会说明他们为什么要这么做，为什么在这个领域鱼和熊掌不能兼得。

对象描述

在自动化测试中，对象泛指一切可能被捕获和虚拟的东西，一个按钮就是一个对象，一组文字也是一个对象，一个程序进程也是一个对象。自动化测试工具的核心就是对象识别容器，其他看到的脚本、时序、数据、外调等都是围绕这个核心来开发的。总体而言，针对Win32的对象识别难度要高于对Html对象的识别，主要原因就是早期Win32的体系设计和微软大量未放开的结构与API声明，所以本文使用Win32对象展开一个综述。

Windows操作系统自带“计算器”程序使用了大量的标准Win32控件，针对其对象识别又涉及到一个很著名的单词“句柄”。句柄是个非常泛意的单词，这又要涉及到Windows系统的一系列机制和相关API函数的调用。我这里简单描述一下，窗体（window）诞生首先要向Windows系统进行注册，注册函数为RegisterClass（API），注册完成后由CreateWindow（API）进行窗体构建，如果这个函数执行成功那么就会返回一个32位的整数值，该值简称为窗体句柄hwnd。窗体句柄值的大小倒是无关紧要，不过它关系着窗体对象，所有需要对窗体进行的操作都必须通过调用它。这个时候，显示屏幕上还不能马上出现窗体，接着使用ShowWindow（API）和UpdateWindow（API）这两个函数后，这个窗体才会最终出现在我们的计算机屏幕上。剩下的就是构建窗体消息循环，在消息回调函数中声明消息处理方式等，所以Windows也会被称为消息驱动的操作系统。句柄在Windows系统中应用频繁，除了HINSTANCE（实例句柄）、HWND（窗口句柄）、HDC（设备句柄），还有HICON（图标句柄）、HCURSOR（光标句柄）、HBRUSH（刷子句柄）等，正是因为句柄的存在，所以自动化测试工具才能如此顺利的操作窗体对象。在Windows系统里可视对象和部分非可视对象都属于窗体范畴，Button也是，List也是，TextBox等都是，窗体即容器，最终导致整个Windows变得如此“多姿多彩”。下面的工具列出“计算器”程序部分被捕获对象：

“计算器”程序中包含的对象呈树形结构排列，描述方式为句柄、窗体名称、类名称，通过捕获描述，基

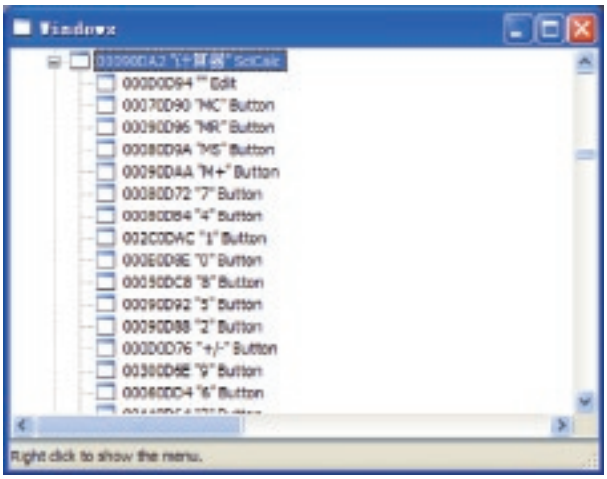


图1 “计算器”程序部分被捕获对象

本上可以很快的定位到实际窗体对象。商业测试工具会提供很多的可选对象属性，原因就是属性越多，对象识别容器识别对象的精度就越高，负面的是消耗的资源也会相应的增加。有的商业工具采用了“映射”机制，即用户所看到的界面显示对象属性并非是对象识别容器内真正的对象属性，是由测试工具在真实对象上层构建的Object Maps。这样处理的好处是降低了用户理解对象的难度，还可以附加更灵活的搜索模式交于用户选择，例如QuickTestPro提供的“属性正则表达式”、“对象Index或者Location定位”等。另外“对象重定义”技术也是采用了上述原理，将非标准对象在UI层面显示为标准Win32对象，在允许范围内使用句柄、消息对实际对象进行控制。

虚拟化对象技术是由非标准对象识别而诞生，其实操作原理为用户操作鼠标对需要进行虚拟化的有效区域画出一个不可视矩阵。随即，对象识别容器关联父对象，记录虚拟坐标值，通过钩子模拟鼠标和键盘消息可以让原来无法捕获的对象也实现正常的人机交互操作（Web的虚拟化对象是挂载在对应的Page父对象容器中）。理论上虚拟化对象无上限个数，但是过多的虚拟化对象会影响测试工具的实际执行效率。

在对象的世界里，并存在着测试对象和实时测试对象（也称运行时对象）的概念，有的测试对象在实际软件运行时相关属性会发生变更，或者有些对象在静态捕获的时候根本不存在，它只存在于程序运行时态，那么这些特性多少都会影响对象识别容器的实际识别效果。所有的商业测试工具中都隐含了一些插值对象属性，以最大的灵活度来匹配、捕获和重构对象，这些都作为识别的核心执行机制而被保护起来，相关的技术文献并不多。在TestMice中，我们也是用了一种特殊的机制来增强运行时对象的识别，尽量做到所见隐藏皆所得，达到识别自动变形的目的。可以说在Win32领域，各家的商业测试工具识别效果都差不多，好在这个Win32时代离普通用户是越来越远了。

在Win32或者Html对象取值中，经常有人问我自动化测试工具是如何实现的。其实只有两种模式而已，第一种即调用对象本身支持的属性通过相应函数获取，例如Win32中使用标准函数getText（API）。第二种就是构建自己的OCR识别程序进行矢量字转换，后者需要投入大量的时间和金钱，当然也可以购买现成的OCR产品，以支持库的方式写入到测试工具中，这些不管是什么测试工具都是支持的。

脚本构建

个人认为，在所有的自动化测试工具中，脚本的组合形式是决定用户是否选用该工具的关键之一，很多测试工具都输在用户对其脚本的第一视觉印象中，其表现形式如下：

```
Start Main()
    SetWindow("Caption=Scical",
"")
    PushButton.Click(
"ObjectIndex=3") //4
    PushButton.Click(
"ObjectIndex=32") //+
    PushButton.Click(
"ObjectIndex=8") //5
    PushButton.Click(
"ObjectIndex=27") //=
    Result = Compare(
CompareProperties,"Text","9.")
//比对结果
End Main
```

可以看出这是个自动化测试脚本，但是至少让我在短时间内看不出脚本与对象之间的关联，这种脚本设计相对比较失败。有的时候，软件行业充斥着“自己好用，用户就好用”的研发思维，过于突出技术而忽略了用户体验。如果这些你都可以快速适应，那么就让我再增加一些脚本量：

```
Start Main()
    SetWindow ("Caption=Scical",
"")
    PushButton.Click(
"ObjectIndex=3") //46
    PushButton.Click(
"ObjectIndex=21")
    PushButton.Click(
"ObjectIndex=32") //+
    PushButton.Click(
"ObjectIndex=8") //52
    PushButton.Click(
"ObjectIndex=2")
    PushButton.Click(
"ObjectIndex=27") //=
    Result = Compare(
CompareProperties, "Text","98.
") //比对结果
End Main
```

上述脚本组成方式，我相信很多实际从事测试行业的人员都是强烈排斥的，这样的脚本越多就会让整个自动化测试项目在重构，分拆和管理上陷入巨大的泥潭，进而加速整个自动化项目的死亡。好的脚本构成应该更贴切用户，让用户能快速观察、快速理解，使用极低的培训成本即可达到快速书写的目的。在TestMice项目中采用一种比较贴切使用者，能快速理解，用极低的培训成本即可达到快速书写测试脚本的方式。并且，这种脚本组成方式便于其维护、分解和重构，例如在对“计算器”程序进行加减乘除运算测试中，TestMice的脚本组成方式为：

```
Window("计算器").Button("4").
click();
/*或者Window("windowtext=计算
器;HWND=234985").Button("4").
click();附加属性匹配
或者Window("windowtext=^计算*").
Button("4").click();附加正则表达式
匹配
或者Window("算").Button("4").
click();模糊匹配
*/
Window("计算器").Button("+").
click();
Window("计算器").Button("5").
click();
Window("计算器").Button("=").
click();
Assert(Window("计算器").Edit(
"").getText().trim("",".").
"9");
```

上面代码中的“或者”分别代表3种增强脚本模式：附加属性匹配、附加正则表达式匹配、模糊匹配，即用户可以在脚本中以自由添加指定属性的方式来增强对可识别对象的操作能力，其中“模糊匹配”为最高等级的容错识别模式。如果是对多数字进行操作，我们还可以自定义函数包容对象操作，例如：

```
public static function
mockNumberDClick(String number)
{
  int i = 1;
  for(;i <= len(number); i++)
  {
    Window("计算器").Button(
right(number,i,1)).click();
  }
}
mockNumberDClick("20");
Window("计算器").Button("+").
click();
mockNumberDClick("20");
Window("计算器").Button("=").
click();
Assert(Window("计算器").Edit(""
).getText().trim(), "40");
```

在我们的脚本定义中，需要达到几点要求：第一，脚本与对象的切合要紧密，尽量做到很高的关联可视化。第二，脚本内带规则检查，尽量在第一时间将用户输入性错误排除，节省时间。第三，脚本有常规函数案例显示，并且支持同步对象集合及内嵌线程扩展。第四，脚本有更开放的接口，方便嵌入模式更好的实现。

测试人员既要实现测试动作逻辑，又要特别关注测试数据的选取，如果可以让测试人员从脚本中就快速的获知对象关联，降低他在这块的开发时间，那么这个交互设计就是成功的。简单的设计即可达到最佳用户体验，这也是TestMice项目一直追求的。

描述性编程

如果对自己的对象识别容器有足够的信心，那么支持“描述性编程”是必然的，这些都是经过长时间的测试和升级而最终被确定下来的。为什么Win32中这种技术未被广大用户所知（例如HP Winrunner中即存在描述性编程技术），是因为Win32对象的非标准性和不确定性造成的，大量对象根本无法被捕获，属性的精确获知就更加不可能了，如此大肆推广描述性编程实在是个不明智的商业之举。进入Web时代后，由于W3C的强力介入让Web有了一个相对统一的标准，所以描述性编程终于发展起来了。

描述性编程的概念就是要使用者直接脱离传统的识别对象-存储对象-再操作对象的模式，升级为你无须进行这些繁琐的操作，直接使用轻量级的Spy工具查到被测试对象的几个关键属性，即可快速的书写测试脚本以加快开发进度，节省出更多的时间用于调试和脚本重构。一般描述性编程脚本结构为：

```
//QuickTestPro中的描述性编程格式
对象(“属性:=‘值’”, “属
性:=‘值’”, “属性:=‘值’”,…
…).动作 (“值”)

//SilkTest中的描述性编程格式
对象(“属性值1|属性值2|属性值3|…
…”).动作(“值”)
```

在TestMice中，这种描述性编程脚本结构微调为：

```
对象(“属性=值;属性=值;属性=值;…
…”).动作(“值”);
```

描述性编程是对对象多属性复合应用技术的集中表现，在性能上也没有太多的额外耗损，如果用得好还可以节省总测试脚本的执行时间，也更加利于调试。所以，不管采用何种描述性编程格式，本质上是没有任何区别的，可以这样说，描述性编程是自动化测试工具一个必然发展的结果，所有的工具厂商都希望使用者更加关注测试数据和测试业务逻辑，脚本本身的表现形式则是越简单、越直白就越好。

测试数据驱动

数据驱动在自动化测试中至少包含了两种概念，一为对象关键字驱动，二为业务关键字驱动，采用什么样的驱动模式完全取决于使用者本身。下面的表格是对Windows计算器程序进行对象关键字驱动的描述：

Window	Obejct	Description	Action
计算器			
SciCalc	Menu	查看，标准型	NULL
	Button	CE	Click
	Button	1	Click
	Button	+	Click
	Button	1	Click
	Button	=	Click
	Edit	NULL	GetText
	NULL	Assert	2
	Button	CE	Click

对象关键字驱动本质上就是对可操作对象做一个显式列表，然后选择一种数据存储方式（文件、数据库、

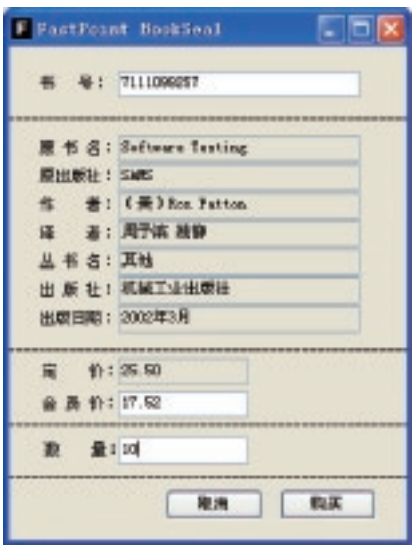


图2 业务关键字驱动表现形式

Excel等）将这些属性、动作存储下来，最后递交给脚本和对象识别容器统一实现。其好处是交互简单明了，坏处是对象变更了整行数据就作废了，造成维护成本偏高，这也是众多商业自动化测试工具的通病。

业务关键字驱动目的性强了一些，所有关键字都跟业务描述和测试数据挂钩，针对图2这样的业务。

其业务关键字驱动表现形式为：

书号	会员价	数量	验证
7111099257	25.50	10	正确
7111255164	78.00	1.5	正确
7111255164	78.00		请输入购买数量
1345266574	49.00	9999	购买数量超过库存数量，请重新输入
3456576574	35.00	0	购买数量输入错误
3456576574	35.00	-1	购买数量输入错误
3456576574	35.00	0.1	购买数量输入错误

对于业务关键字驱动来说，业务本身的数据和对应验证是最为重要的，只有通过更多的测试数据来达到一个满意的、足够安全的测试覆盖率。

场景恢复

自动化测试执行过程中遇到中断事件是非常普遍和令人沮丧的，我们经常告诫测试人员在脚本的执行机器上不要安装过多的额外软件，就是因为某些软件可能产生一些莫名其妙的对话框或者发送一些古怪的消息来阻断正常自动化测试进程。那么作为一款强大的自动化测试工具，保存这种错误场景和继续恢复测试正常执行是非常必要的，这些统称为场景恢复。在场景中，我们需要分析哪些现象会阻断脚本的正常运行？或许是一个软件升级的对话框，或许是来自于网络的一个弹出式窗体，或许是防火墙拦截到的一段信息提示等等，这些一切皆有可能。那么如何处理这些令人讨厌的事情呢？ TestMice项目中提供3种场景恢复，即：

- Windows弹出式对话框场景恢复
- 被测试软件错误场景恢复
- 对象属性变化场景恢复

第一种场景还是需要被动的预先获知弹出式对话的句柄，或者通过预设对话框Title属性，一旦该机制被触发，测试软件会自动寻找符合Title属性的窗体，然后启动由用户选择的4类后续对应处理：

- 模拟鼠标和键盘消息，进行关闭窗体或者其他可定义操作。
- 关闭阻碍自动化测试过程的软件进程，可由用户设定，此项较危险。
- 调用预约脚本，例如发送紧急短信或者邮件通知等。
- 重启Windows系统，这里建议自动化测试做在虚拟机里，可用虚拟机SDK作辅助插件开发，保存当前软件执行Cookie。

第二种场景在测试过程中是最为常见的，例如已识别菜单项没找到（Winrunner低版本中这种现象比较常见），List下拉项目中数据和脚本预设数据不匹配，出现多个重复的对象（例如打开两个以上相同的IE窗口），或者被测试对象状态为Visable等，所以针对此场景，最好设置一个All Error处理，即不管出现了什么被测试软件错误，一概调用4类后续对应处理。更好的办法是，错误可以由使用者自定义，那么就需要测试工具在实现机制上有更多的考虑了。

第三种场景，对象的属性变化处理最为复杂，不过这种处理机制也是对象识别容器的重要功能，可以由软件内部提供对象重新模糊匹配方式，略过该测试案例执行，直接跳转到其他待测试案例执行等。同样该场景的触发也可以调用4类后续对应处理。既然测试执行过程中产生阻碍事件是无法预料的，那么有备无患的后续处理机制就显得格外重要了。

这三种场景恢复TestMice全部支持，并且我们将场景后续处理机制做在了插件里，这样的话就把对其他场景的定义交给了实际使用者，因为我们永远相信，预设的都是不够的，是最容易招惹抱怨的。

插件和编译运行

目前很多测试工具有限的放开了二次开发接口，可以让使用者进行受一定限制的功能开发，但是至今还是没有一款商业测试工具放开到可以操作其对象识别容器的地步。TestMice完全提供操作其对象识别容器内核所用的SDK开发包，以供任何个人或者企业制作具有自己特色的自动化测试应用。另外，我在大量的自动化测试项目实施中发现了一些很有趣的事情，即目前市面上的测试工具脚本大多是以明码方式运行的，企业比较抵触这种方式，认为这种明码执行方式有泄漏企业秘密的重大嫌疑。那么在TestMice项目中有个插件专门负责把脚本编译成二进制模式，顺便对脚本进行优化和执行。插件是所有商业工具扩展的灵魂，开发商如果没有足够的想象力，那么就把这项任务交给广大插件开发者来完成吧。

总结

国内至今尚未有正式机构对自动化测试工具进行整体式的深入研究和商业研发，自2007年国内某人开始制作成体系的研究性自动化测试工具，直至今日发展颇为艰难。在经历了一个又一个的技术攻关后，终于让我们开始成熟起来。项目组同样希望TestMice可以成为国内每个测试人员，甚至是每个开发人员手中的测试利器，国产的商业自动化测试工具应用年代终于离我们越来越近了！

译者简介



王东刚，曾出版《软件测试和JUNIT应用》。2006年创办杭州日辉信息技术有限公司，专业提供软件测试咨询及实施业务。个人评论站：www.fastpoint.cn。

责任编辑：郑柯（zhengke@csdn.net）

没有银弹，量体裁衣

——Visual Studio Team Architect 团队的敏捷软件开发之一

■ 文 / Ramesh Rajagopal　译 / 钱量　朱永泰　陆榕　林裕科

在最近几次与客户面对面的交流中，我有幸分享了我们团队如何在日常工作中进行敏捷软件开发。毫无疑问，这在中国开发人员中是个热门话题，我也希望通过《程序员》与更多读者进行书面的交流。当然关于敏捷开发利弊得失的争论有不少，而相关的开发模式也分成了TDD (Test Driven Development)，Scrum，XP(eXtreme Programming)等流派。就我个人而言，一个团队是否严格遵循某种既定的敏捷方法并不重要，但一定得选择并采用一种（或几种）最适合自己开发团队和开发项目的。我认为重要的是团队能否遵循《敏捷软件开发宣言》所涉及的12条原则。

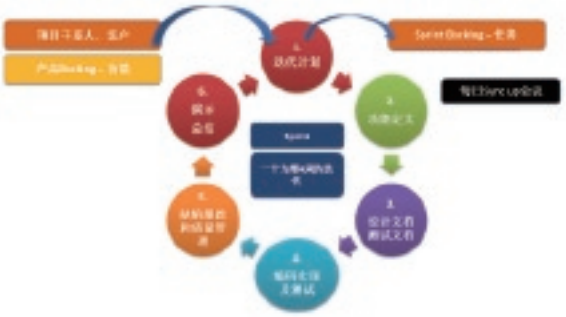
在我深入这一议题前，请允许我介绍一下团队：我们属于微软开发工具部（Developer Division，以下简称DevDiv），这个部门拥有几千名软件工程师，核心产品Visual Studio系列的用户从软件开发爱好者一直到大型企业里的专业开发人员及架构师。

大量而且复杂的依赖关系、代码改动、紧迫的开发周期等因素使管理软件开发生命周期并按时发布高质量的Visual Studio产品极具挑战性。为了降低风险和复杂度，DevDiv在开发Visual Studio 2008过程中采用了功能分支架构(Feature Branch Structure)和功能小组模型(Feature Crew Model)。其实这一方式之前已在Office开发团队的实践中取得不错的效果。它的最大好处之一就是使负责某个功能的团队在独立开发过程中有更大自由。在工具方面，我们使用微软的Visual Studio团队套件，充分利用这个丰富的工具集使团队的每个成员各司其职同时又保持高效协作。这套工具配合我们采用的敏捷开发流程使用，在实践中体验了卓越的价值。由于篇幅所限，在这篇博文中我将侧重介绍我们团队是如何进行敏捷软件开发的，之后我将为大家介绍我们是如何在日常工作中使用这套Visual Studio团队套件的。我们团队负责Visual Studio系列中的Visual Studio Team System Architecture Edition，帮助架构师、运营经理及开发人员以可视化方式构造面向服务的解决方案、降低（软件产品开发的）复杂度。目前我们已开发了基于UML和DSL几个建模工具。这基本上是一个全新项目。

从产品开发来看，我们属于全球分布式开发，团队分布在三大洲的四个城市，包括亚洲的上海，北美洲的雷德蒙和夏威夷，以及欧洲的剑桥。为了尽可能减少分布式研发对团队间交流所造成的障碍，我们尽量使功能小组的成员集中于一地。

在本文后半部分和之后的文章中，我所谈及的敏捷软件开发流程都是同一个功能小组所遵循的，即是我们中国团队所遵循的。

我们中国团队主要负责开发基于UML的核心图形设计工具，包括即将发布的Logical Class Designer，Use Case Designer。我们所采用的敏捷开发方法是Scrum的修改版。就如我之前提到的，我们认为敏捷开发方法和技术没有哪一种是万灵丹，适合自己才是最好的。下面是一个我们敏捷软件开发流程的概要视图：



敏捷软件开发流程的概要视图

产品待开发事项（Product Backlog，视图的左上角）可以被视作一份这个团队以优先级排列的、需要完成的功能需求单：来自相关产品利益相关者（Stakeholders）对产品提出一系列高端要求。例如，我们最初的要求是为客户增加逻辑级（更抽象的）和物理级（更靠近代码）建模提供支持，由此

衍生出高端功能需求，诸如开发在逻辑级方便客户生成逻辑模型、兼容UML的关系图和开发帮助创建无力模型的DSL关系等。然后我们会对将要支持的UML关系图种类按优先级进一步分解（UML共有13种不同的关系图）。产品利益相关者的意见会驱动整个优先级选择过程，最终我们得出五个最重要的关系图：Logical Class Diagrams，Use Case Diagrams，Sequence Diagrams，Activity Diagrams 和 Component Diagrams。于是，团队依据当时对产品和市场的了解，以故事标题的形式完成一份产品待开发事项。无疑，整个开发工程中一旦要求发生变化，也会导致需求排列优先级的变更。

在与客户的交流中，我被问得最多的问题之一是是否需要在敏捷开发过程中创建架构模型设计。和咨询公司一样，我的答复也是：视情况而定。围绕Big Design Upfront (BDUF)，发 You Are Not Going to Need It (YAGNI)以及让团队在开始实施新功能时“重构”现有的代码/设计等所存在陷阱的争论也不少，其中有不少值得借鉴。尽管如此，我坚信设计初期存在这么一个阶段可以尽责地做架构设计以生成高端架构。例如，你打算建一个网上贷款流程的应用程序，你可能需要决定在这个架构里有几层。当然，能有这样一个基于最初要求的，并可能随着项目进展有所变更的架构是很重要的。在我看来，重构在敏捷开发中有其重要地位，但是如果是变更基础架构的“大重构”代价就太大了。

在我们团队所遵循的敏捷软件开发实践过程中，我们的项目被分解成类似Scrum的若干个四周sprints或迭代开发周期。尽管没有测试驱动开发(Test Driven Development)或结对编程(Pair Programming)，但我们的开发人员会编写单元或签入测试（Unit/Check-in Test）来检查功能，开发和测试工程师也会在一起调试、调查或评审某个特定问题和变更等。我们还会使用极限编程（eXtreme Programming）中的用户故事（User Story）模式。事实上，我们的产品待开发事项和每个迭代周期中的待开发事项（Sprint Backlog）都会以故事的形式被追溯。这些用户故事就是描述一个系统的最终用户会如何使用某个特定功能。

下面我将和大家具体探讨一下我们是如何进行Sprint计划的。通常，我们都会在一个Sprint阶段的最后一周计划下一个Sprint阶段。首先，我想说的是，不存在敏捷无须计划的神话。可是，敏捷开发中的计划的确和传统软件开发中的计划有着很大区别。我们针对利益相关者给出的上层需求创建了带有优先级的产品待开发事项(product backlog)。这一带有优先级的任务列表形成了最基本的sprint计划。在这一过程中，我们一般遵循三阶段的步骤：在主管间进行的预计划阶段，所有团队成员都参加的计划阶段以及包含利益相关者的计划提交阶段。这里的关键是：计划是在所有成员的通力合作中进行，最重要的是由组员自发来制定标准、而不是依赖于某个项目经理。

预计划阶段在上一个sprint的最后一周进行，在这一阶段中，团队中分别带领项目经理，开发和测试的几位主管会聚集到一起讨论出在即将进行的下一个sprint中，需要开发的故事(story)列表。这个过程取决于很多因素，其中最重要的是：上一个sprint的进展情况，从利益相关者那里得到的反馈，需求或故事优先级发生的变化以及预计的团队速度。项目经理(有时甚至是开发人员或者测试人员)在阐述故事的时候会尽量简短到只描述出目标、故事的简单介绍以及故事的具体流程。我们发现OneNote很好的满足了我们这一需求。

产品待开发事项总是列出对客户有价值的条目，同时它也可以增加这个团队要求的条目。但是，只有那些最终会给客户带去价值的条目才可以出现在待开发事项中。举例来说，创建并维护一个持续集成服务器以持续保证最终产品的质量，这样的条目被允许出现在待开发事项中的。

计划阶段通常在sprint的第一天进行。在开会前，项目经理会把OneNote页面的链接发送给组员，以便大家评估，并且为计划会议做好准备。通常，组员会在OneNote页面中交换意见，从而在会议之前澄清那些不明了的地方。在计划会议当天，团队组员会聚集到一起，过一下所有的故事，解决之前发现的任何问题，把故事进一步细分成一些任务，并描述每个故事的验收测试。组员同时也会对完成这些故事所需要的时间做一个大致的估计，然后根据这些估计决定在这个sprint中，团队可以完成哪些故事。

计划提交阶段在之后的一天进行，主管会再度聚集在一起并且向利益相关者介绍团队承诺完成的任务。此时，利益相关者可以提出建议对优先级进行调整。比如，如果团队成员可以完成故事A，B以及C，但是不能完成D和E，利益相关者可以建议团队在这一个sprint中完成A，B，D以及E(假设D和E消耗的总时间和C相同)。然后，项目经理会把这些故事输入用来管理我们项目的Visual Studio Team Foundation Server。

注：我们花了好几个sprint来学习并总结出以上这个计划流程。这就是sprint回顾发挥的重要作用。

作者简介



Ramesh Rajagopal是微软中国研发集团服务器与开发工具事业部的部门经理，在上海带领一支团队开发Visual Studio Team System中的UML建模工具，并多次受邀与多家独立软件开发商分享微软敏捷开发的经验。Ramesh Rajagopal的大部分职业生涯致力于包括Visual Studio在内的软件开发工具研发。Ramesh Rajagopal毕业于北卡罗来纳州立大学并获计算机科学硕士学位。

■ 责任编辑：郑柯 (zhengke@csdn.net)

实现用户的三个情感诉求

——QQ2009用户体验设计谈

■ 记者 / 圣伟

回顾腾讯的整个IM平台发展过程，总共可以分为三个时代：第一个时代是从1999年到2001年，当时中国的互联网正处于萌芽时期。以QQ第一个版本为代表的即时通讯工具的出现，解决了用户跨空间沟通的障碍，而QQ在当时的主要功能还仅仅是以沟通为主的最“朴实”应用；第二个时代是从2002年到2006年，时值互联网行业横跨从泡沫破裂到全面复兴的阶段。在此阶段，中国网民的需求呈现出多样化态势，这时的QQ则主要强化其商业价值，从单一的IM发展成了一个互联网平台——一个实现商业模式的即时通讯平台；而QQ2009的发布，则象征着第三个时代的到来，更多地体现了一种用户价值的回归。

以QQ2009为代表的第三代IM平台，除了在软件架构上有新突破外，在用户体验方面，也实现了基础通讯的全面优化和新功能的设计推出。



以用户需求为导向的QQ项目团队

QQ2009的理念和应用环境

贯穿整个QQ2009平台的理念，可以总结为三个字：“轻、薄、快”。QQ2009拥有一个跨平台、多场景的全应用环境。首先，QQ2009基于Hummer平台，能全面兼容、支持Windows/Vista/Linux/Mac等多个系统平台，更可在PC、手机与无线终端之间切换。

其次，用户还可以根据需要，在商务、家庭和学校等多个场景间变换，选择不同的功能支持。

第三是定制化和模块化。用户可根据自己所处的场景和终端需求，对QQ定制化操作，不同场景的功能切换和安装，QQ均可实现模块化操作。腾讯CTO熊明华在接受采访时介绍说，以QQ秀的组件升级为例，过去可能需要重新发布一个版本，而在QQ2009模式下，只需要升级相应的模块就可以，比过去的开发时间节省了近三分之二。

第四是集成社会关系管理。熊明华表示，QQ作为一个人际关系管理软件，承载着人们的各种社会关系。由于好友关系链的存在，使得所有使用QQ的人，实际上都是存在于一个大的社会化网络中，QQ2009将SNS的功能进一步强化，支持用户自主选择的权利，能够有选择地对自己的社会关系链进行实名或匿名管理。

第五是全面的国际化。第六则是安全性能的提升，提供了更全面的安全保障和服务，包括账号体系、聊天记录加密体系和信息过滤体系的全方位安全生态建设。

追寻用户的潜在情感诉求

除去对初期设计理念的坚持和应用环境的量化及性能考量，体察客户需求，甚至寻找其发自内心的情感诉求，从而指导用户体验设计，这也是QQ2009研发团队着力解决的部分。

第一步，产品经理通过论坛、博客，直接与用户接触，挖掘用户的原始诉求。根据这些用户诉求，再结合现有的运营数据进行分析。对于大众的诉求，腾讯会尽快讨论产品概念，然后形成产品需求。在产品设计过程中，通常是产品经理、交互设计师、开发人员在一起，根据产品概念来设计产品，这是一个团队成员共同参与的过程。此外，腾讯还有强大的用户研究团队，从产品概念产生到产品形态形成的过程中，该团队都会协助产品部门一起，使最终的产品形态高保真地切合原始诉求。

在QQ项目团队的心目中，用户有很多维度：每个团队成员自身就是一个很挑剔的用户，其次周围的同事，甚至其老板、亲友，都是最好的用户。多年来，QQ项目团队总结出几个心得体会：

（1）用户的需求多种多样，需要挖掘。同一功能，不同用户感受截然不同，需要放在环境中来看，要充分辨析want和need。

（2）海量用户的产品，所设计的功能并非为了满足所有海量用户。工作中

很大一部分就是在肯定和否定中寻找那些被大部分人肯定或接受的东西，每一个功能都有特定的使用人群，真正能满足到核心用户，这是最大的挑战，如果能做到，就能得到最大、最好的口碑。

（3）充分尊重每一位站出来发表意见的用户。聆听他们的声音，并耐心追问再追问，因为他们是最好的测试“同事”。在QQ2009设计过程中，有一个典型案例：一个测试用户在体验2009之后，对征求他意见的测试人员提出疑惑——2009为什么把08的消息盒子去掉了？其实，在主面板上这个功能是留有快捷键的，但是用户不能直接发现。问题找到了，接下来要做的事情自然就很明确。

（4）用户用的是产品，不是说明书。在做各种新旧功能时，方便、快捷、简单、显而易见始终是最重要的，用户很少关心新产品的新增功能文档。“使用前请阅读产品说明书”这样的话，绝对不是网络时代的东西。互联网时代，用户对新产品所做的第一件事是使用，第二件乃至最后一件，还是使用再使用。有一个例子：用户说她很喜欢系统设置中的皮肤设置功能，但是入口太深。经过沟通，才明白她不知道有快捷键存在。这提醒我们，能提供用户需要的东西，却不一定能让用户感觉到并且使用上，这也是一个大问题。对用户来说，简单、直接是最好。

（5）让用户参与到设计中。很多时候产品部门马不停蹄地进行设计，往往会陷进去无法自拔；无法知道真实的世界是怎样的，用户真正想要的是什么，以所谓专家经验来推导或者模拟用户，这始终有着极大的局限性。要让用户参与进来，倾听用户的心声，帮助设计者自己“跳出来”思考，更客观地面对自己的设计，才能做出用户真正需要的产品。

通过这一套较为成熟的方法论，腾讯QQ项目团队挖掘出了分别对应QQ用户的三个潜在情感诉求——“自由自在”、“爱与尊重”和“操控的欲望”。

实现用户的三个情感诉求

理清楚用户的三个诉求后，QQ项目团队就开始了实际行动。

用户自定义的界面选择——让用户“自由自在”

QQ2009强化了个性化的使用体验，得益于第三代IM平台Hummer软件架构的支持，QQ2009在调色、底纹、换肤、头像自定义等方面做了全新的交互设计和视觉优化，也带起了腾讯其它产品系列相关特性的潮流。

保护用户隐私——给用户“爱与尊重”

以QQ锁为例，QQ2009设计师在通过不同渠道了解用户想法的时候，很多用户提到在特定场景时想要隐藏自己的QQ。为满足用户需求，团队快速进行了头脑风暴，大家很快想到类似QQ锁的功能，并进行原型设计，很快就有了初步方案，并通过进一步的用户测试和论坛反馈进行着持续的优化。



图1 QQ2009中的QQ锁界面图

“QQ锁”可以有效地保护用户隐私，当用户离开电脑的时候，“QQ锁”可以保证用户的聊天信息不被偷看。以前，用户要想阻止别人偷看自己的隐私，只能将QQ关掉。而现在，只要使用“QQ锁”将QQ锁定就可以，同时QQ仍保持在线状态，不管用户离开多久，都不会错过任何信息；而正在接收或发送的文件也可以继续传输不会中断。另一方面，当QQ被锁定之后，会自动断开与音视频或者远程协助的链接，避免用户的隐私在无意中被泄露。此外，QQ被锁定之后，所有的声音也被关闭，这样即使有好友给你发送了信息，正在使用你电脑的人也不会知道。通过这种办法，用户各方面的隐私都受到了保护。



图2 QQ2009的消息盒子

对于以后的改进，QQ项目团队设计人员还将从更安全、更人性化的角度来继续发展这一功能：如从安全角度考虑，当QQ被锁定后，将禁止在网页上的“一键登录”功能；而从人性化的角度考虑，可能会为“QQ锁”增加一个独立密码，让用户在解锁的时候更便捷。

异步消息盒子——满足用户“操控的欲望”

消息盒子是QQ上统一收取各类信息并支持异步处理的容器，新版消息盒子带来了一系列的贴心设置：改进的消息列表使用户可以快速浏览到更多信息；独立的工具栏操作更方便，支持拖拽更自由；新增了预览视图功能，已读、未读消息一目了然。此外，它和业界其他类似产品最大的不同，就是能将即时消息和异步消息一网打尽，用户可以在消息盒子中集中挑选自己所关心的信息做快速阅读。

未来规划

在接受本刊采访时，腾讯QQ2009项目团队成员表示，未来版本的QQ用户体验设计中，除了将继续加强基础通讯、好友管理方面的体验优化和功能创新外，还将进一步基于已有技术做进一步预研和优化。相信在不久的将来，QQ项目团队还将实现更多超乎用户预期的特色功能。

一分钟先生
Mr. One Minute

如何应对客户方的人员变更?

问题：在软件项目中，客户方面的人员变更会给整个项目造成很大影响，有时客户方领导层的变更甚至会导致整个项目之前的工作前功尽弃，最极端时还会出现取消的状况，那么如何管理和应对这个问题，才能保证将其负面影响降到最低？



Tom Ticknor
科维安（QAI）全球总部首席运营官

这个话题涉及到整个软件项目管理。根据我的经验，由于客户人员变动导致的项目变更或取消所引发的问题，大约只有25%的因素是跟编程人员有关的。因此在一定程度上可以说，这个问题更多应该是分析团队和项目经理们应考虑的范畴。管理团队要在项目前期做出充分的考虑，如对整个软件开发流程做出规范，包括与客户沟通的规范、项目管理的规范，以及可能出现的客户人员变动时的应对情况等。

一旦客户方出现人员变更，项目经理应该在第一时间获知客户相关人员变动的真实情况，并作出及时反馈。采用敏捷方法的原则可以较好地处理这个问题。在敏捷模式下，计划可以不断调整，开发团队通过阶段性的工作，将开发成果交付到客户手上，并获得最新反馈以便及时做出应对。这些方法，都可以将上述情况出现时的损失降到最低点。



阿朱(吕建伟)
CTO，《走出软件作坊》图书作者

中国自古有句老话“一朝天子一朝臣”，所以客户方领导层的变更往往是一个软件项目的大灾难。一旦发生领导层变更，作为乙方的IT公司还无法插手更多，毕竟这是客户自己内部的事情。在这种情况下，最怕的是新的领导迟迟无法选定，那所有项目都会因此搁置，众人都在观望风向。但是，我们也还需要积极地行动：

- 深入了解。多方收集客户方项目领导层换掉的原因。从换掉原因中推测更高领导层的想法，再推测自己手中项目的可能命运。摸准更高领导层的想法，调整项目重心。
- 小步快跑。分析出自己在这个项目中最精要最能给客户短期内带来经济价值的部分，游说客户方项目执行管理者项目不要停下来，小步实施，小范围实施，一点点推进。
- 定期汇报。把自己目前做的事情，已经做了什么，产生了什么明显价值，下一步还有什么非常值得做的，定期汇报，而且间隔较短的时间内和客户方保持联系，不断巩固客户信心。
- 加深应用。不断深入培训、加强积极的服务支持，把最终用户的使用夯实。一旦用户习惯了现有这种工作模式，再让其改变就会有很多不适应。新领导变革的时候也会考虑到过去的承接、变更成本、变更难度、变更时间。





王晓明
组织转型专家、敏捷教练和商务分析师

首先分析这个问题产生的原因。新的客户领导之所以改变项目方向，改变想法，甚至取消项目，主要基于以下几个原因：

（1）项目目标在开始阶段没有清晰确定。

（2）项目没有清晰的ROI分析。在项目进行到某个阶段的时候，客户看不到项目所能带来的商业价值。

（3）项目质量不能满足客户需求。

（4）其他商业方面的因素。

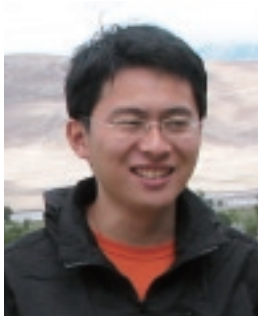
针对上述原因，可以尝试以下解决方法：

（1）项目初期，整个项目团队和客户所有项目相关方共同确定目标，并在项目过程中不断检验，确定项目团队的每个成员都对项目目标有清晰的理解。

（2）清晰正确地描述每个Story或者很小的功能点的商业价值，如果能将其转化成现金最好。确定项目总是将高商业价值的Story或者功能点赋予高优先级。

（3）对项目质量的保证，敏捷中的很多实践都是有效的，比如TDD（测试驱动开发）、自动化测试、及时的交流、合理的重构等等。

（4）如果一个项目做到没有偏离目标，能不断地给客户带来利润，且有较好的质量，新客户领导还不满意，这就要去面对面的交流是不是有其它原因。比如客户自身业务运营的问题等，看看你们是否能帮助客户解决他的问题。



金毅
项目经理，多年来服务于欧美软件外包行业，从事管理工作

我建议把客户人员变更看成一种风险来管理，所以要谈应对，主要就从发生的概率和影响来着手管理。

降低概率。只有了解客户的情况，才能大概清楚是不是会有人员变更的可能。所以可以在与客户的常规例会中增加议程来聊聊两边的情况。同事中如果有Onsite的人员，需要提高这方面的意识和敏感度，尽早把情况反馈回来。

降低影响。客户人员变更带来的常规影响主要有：

（1）可能有需求变更导致设计、代码等返工，由此带来工期、工时等问题。为了降低这种影响，短迭代、尽早交付是个不错的选择。对于固定价格合同的项目，需要预先定义好相应的变更机制来处理这种情况，在估算的时候，需要多设置假设条件来约束客户。当然很多时候，光靠Sign off是不够的。可能的话，尽量采用TM的合同类型。

（2）新客户可能对原先工作突然不满。新官上任，总是要点点火的。也许原先客户负责人对项目进展还算满意，但换个人来看什么都不爽。这时的关键点就要一开始定义清楚验收标准和相关指标，并且做好相关数据的收集。也许客户不需要这些，但这却是一种很好的自我保护措施。究竟什么样的指标需要定义呢？绝大部分都是很日常的，比如工时分布、缺陷率、代码注释率、代码质量、测试覆盖率、客户反馈，等等。有了这些数据，新负责人就没那么容易找碴儿了。

（3）新客户可能取消项目。要想避免取消项目，一个方法就是尽早交付，来实现/体现客户的市场价值。老是没有可见的东西出来，项目被砍掉的概率就大多了。当然项目取消也不一定是不好的事情。

（4）新客户可能用这个项目作为“政治素材”。这时，项目组的成员需要小心，不得随便评价和原先客户的合作，包括提供项目数据等，而需要把皮球踢给项目经理来处理。

总体来说，客户高层变更虽然风险概率小，但影响会很大。还是得小心再小心。

用三叉搜索树实现高效率的“自动完成”

自动完成能提高网站的用户体验，但要高效率地实现这功能，却是一种挑战。

■ 文 / Igor Ostrovsky

在过去几年里，“自动完成”功能已经在网上遍地开花。Facebook、YouTube、Google、Bing、MSDN、LinkedIn……众多网站都设法在你打字的时候，随时帮你补上完整地词句。

自动完成的确能带来绝佳的用户体验，但要高效率地实现这功能，却是一种挑战。高效的实现往往牵涉到一些有意思的算法和数据结构。本文就将介绍一种可用来实现自动完成的简单数据结构：三叉搜索树。

Trie：简单但空间效率低下

讨论三叉搜索树之前，我们先来看一种简单的数据结构——Trie。用这种数据结构，自动完成所需的查找可以执行得非常快，但需要消耗非常多的内存。Trie是一种形似树的数据结构，它的每个节点都包含一个指针数组，其中每一个指针对应着字母表里的一个字母。从根节点开始，我们只要依次找到目标单词里下一个字母对应的指针，就可以一步步追查到目标。

用C#可以这样实现节点：

```
class TrieNode {
  public const int ALPHABET_SIZE = 26;
  public TrieNode[] m_pointers =
    new   TrieNode[ALPHABET_SIZE];
  public bool m_endsString = false;
}
```

图1的Trie里保存了AB、ABBA、ABCD和BCD这几个词。单词结尾所在的节点标为黄色。

用Trie实现自动完成很容易。我们只要跟随着指针，就能去到与用户当前输入对应的节点。从该节点往下遍历，我们就能列出所有可以补完当前用户输入的字符串。

然而从图1可以看到，Trie有一样重要的缺点。这一页纸之所以能放得下图1，仅仅是因为它只用了{A,B,C,D}四个字母。如果要支持26个英文字母，每个节点就要保存26个指针。假若我们还要支持国际字符、标点符号、区分大小写，内存用量就会急剧上升，以至于不可行。

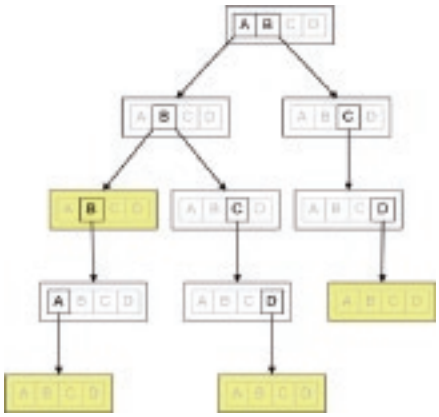


图1 Trie的例子

节点数组中保存的空指针占用了太多内存，我们遇到的困难与此有关，因此可以考虑改用其他数据结构去代替，比如用hash map。然而，管理成千上万个hash map肯定也不是什么好主意，所以我们还是去看看另一种更好解法吧。

三叉搜索树

三叉搜索树用了一种聪明的手段去解决Trie的内存问题。为了避免多余的指针占用内存，每个Trie节点不再用数组来表示，而是表示成“树中有树”。Trie节点里每个非空指针都会在三叉搜索树里得到属于它自己的节点。

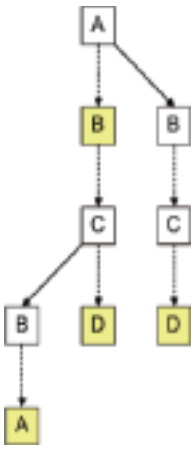


图2 三叉搜索树的例子

例如前面例子中的Trie，在三叉搜索树中会表示成图2的样子

三叉搜索树包含三种类型的箭头。第一种箭头和Trie里的箭头是一样的，也就是图2里画成虚线的向下的箭头。沿着向下箭头行进，就意味着“匹配上”了箭头起始端的字符。如果当前字符与我们期望看到的字符不一致的时候，才会沿着向左或向右

的箭头行进。如果按照字母表，我们要找的字符应该排在当前节点上字符的前面，我们就取左箭头；反之则取右箭头。

例如，图3的绿色路径就说明了我们如何确认三叉树中存在字符串“ABBA”。图4说明我们找不到包含“ABD”的字符串。

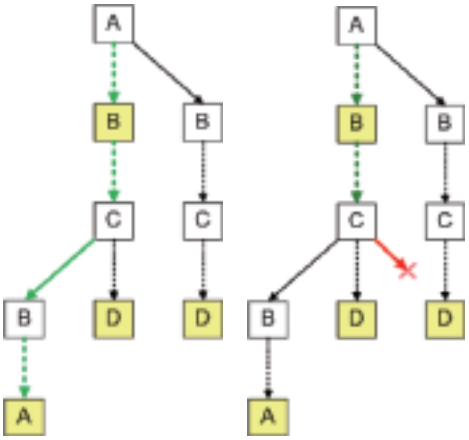


图3 查找“ABBA”的路径

图4 找不到“ABD”

服务器端的三叉搜索树

对于Web应用，自动完成的繁重处理工作绝大部分要交给服务器去完成。很多时候，自动完成的备选项数目巨大，不适宜一下子全都下载到客户端。相反，三叉树是保存在服务器上的，客户端把用户已经输入的单词前缀送到服务器上作查询，然后服务器返回候选单词的一个列表（图5）。



图5 客户端向服务器查询以“bin”开头的词，服务器返回候选单词

实现

以下是用C#编写的一个三叉搜索树的简单实现：

```
public class TernaryTree {
  private Node m_root = null;

  private void Add(string s, int pos, ref Node node){
    if (node == null) {
      node = new Node(s[pos], false); }
    if (s[pos] < node.m_char) {
      Add(s, pos, ref node.m_left); }
    else if (s[pos] > node.m_char) {
      Add(s, pos, ref node.m_right); }
    else {
      if (pos + 1 == s.Length) {
        node.m_wordEnd = true; }
      else {
        Add(s, pos + 1, ref node.m_center); }
    }
  }

  public void Add(string s) {
    if (s == null || s == "")
      throw new ArgumentException();
    Add(s, 0, ref m_root);
  }

  public bool Contains(string s) {
    if (s == null || s == "")
      throw new ArgumentException();

    int pos = 0;
    Node node = m_root;
    while (node != null) {
      int cmp = s[pos] - node.m_char;
      if (s[pos] < node.m_char) {
        node = node.m_left; }
      else if (s[pos] > node.m_char) {
        node = node.m_right; }
      else {
        if (++pos == s.Length)
          return node.m_wordEnd;
        node = node.m_center;
      }
    }
    return false;
  }
}

class Node {
  internal char m_char;
  internal Node m_left, m_center, m_right;
  internal bool m_wordEnd;

  public Node(char ch, bool wordEnd) {
    m_char = ch;
    m_wordEnd = wordEnd;
  }
}
```

注意事项

为了取得最佳性能，字符串应该以随机的顺序插入到三叉树中，尤其不应该按字母顺序插入。否则对应于单个Trie节点的子树会退化成链表，极大地增加查找成本。当然我们还可以采用更复杂的办法去实现自平衡的三叉树。

另外，记住不要超出实际需求而采用华丽的数据结构。如果候选单词的数量相对较少（比如只有几百个），暴力搜索已经足够快了。

作者简介



Igor Ostrovsky，资深软件工程师。目前在微软平行计算平台团队从事软件开发。他的个人网站为www.igoro.com。

责任编辑：郭晓刚 (guoxg@csdn.net)

Transact-SQL元编程

——用T-SQL生成T-SQL

在执行过程中动态生成简洁的SQL脚本，代替设计时硬编码方式的脚本编写，使相同的脚本可以不经修改应用到完全不同的场合。

■ 文 / 王小鉴

概观SQL Server元数据

SQL Server系统目录描述了数据库的结构和各个表、视图、索引、存储过程、触发器的定义。这类信息也称作数据库的元数据。熟悉对元数据的访问，可以提高数据库管理效率，提升管理的自动化水平。假定一个包含人力资源信息的数据库在使用一段时间后，设计者发现表示雇员姓名的字段长度不够，无法存储少数民族雇员和外籍雇员，于是决定扩充字段长度。不幸的是，数据库中有不少数据表都包含雇员姓名字段，一个一个修改起来既麻烦，又很容易遗漏。如果能同时列出所有这些需要修改的字段，那该多好。如果一个SQL语句就能扩充所有这些字段，那不是更好！

SQL Server目录视图体系

SQL Server提供多种方式访问其系统目录。如果是程序员，使用ODBC或OLEDB/ADO方式均可得到SQL Server的结构。单是就采用SQL脚本方式而言，也可以通过访问目录视图、信息架构视图、系统存储过程、元数据函数等途径，其中推荐的方式是目录视图。

SQL Server的目录视图包含若干个只读的视图，可以像访问普通的视图一样获取其中的信息。通常每个重要的数据库概念都有相应的目录视图与之对应：

概念	视图
表	sys.tables
视图	sys.views
表、视图中的列	sys.columns
索引	sys.indexes
主键、唯一约束	sys.key_constraints
外键约束	sys.foreign_keys
CHECK约束	sys.check_constraints
默认	sys.default_constraints
触发器	sys.triggers

在SQL Server的设计理念中，表、视图、存储过程、用户自定义函数、触发器和各种约束均视为一种“对象”。在sys.objects视图中，可以看到这些对象的“全家福”，其中type和type_desc列标注对象的类型。限于篇幅，这里不能一一介绍各目录视图的含义和用法，试举几个例子略作说明。

例如，下面的SQL语句可统计当前数据库中用户自定义函数的数量：

```
SELECT COUNT(*) FROM sys.objects WHERE type IN
('FN','IF','TF')
```

下面的SQL语句查询表t1各个列的名称：

```
SELECT C.name column_name
FROM sys.columns C
INNER JOIN sys.tables T ON C.object_id=T.object_id
WHERE T.name='t1'
```

下面的SQL语句检查t1表的c1列上创建了哪些主键、唯一约束或索引：

```
-- index name or primary key/unique constraint list
SELECT I.name index_name
FROM sys.index_columns IC
INNER JOIN sys.indexes I
ON IC.object_id=I.object_id AND IC.index_id=I.index_id
INNER JOIN sys.tables T
ON I.object_id=T.object_id
INNER JOIN sys.columns C
ON IC.object_id=C.object_id AND IC.column_id=C.column_id
WHERE T.name='t1' AND C.name='c1'
```

目录视图与元数据函数结合使用

从上面的例子可以看出，从目录视图中获取信息常常需要用对象ID在不同的视图之间关联。很多时候，关联只是希望

获取对象名称而已。这种情况下，使用元数据函数OBJECT_NAME直接将对象ID转换为名称要方便得多。如下面的SQL语句不需要关联sys.tables视图就可以列出每个表的主键：

```
SELECT OBJECT_NAME(parent_object_id) table_name,
name primary_key_name
FROM sys.key_constraints
WHERE type='PK'
```

另外一个元数据函数OBJECT_ID则由对象名称反查对象ID，这个函数更常用的一个功能是根据返回值是否为NULL来判断对象是否存在。注意如果考虑架构（schema）的话，则OBJECT_NAME与OBJECT_ID不是完全可逆的，OBJECT_NAME返回值中不包含名称中的架构部分。

与对象的使用类似，列ID的使用也远比列名称频繁。元数据函数COL_NAME负责把列ID转换为列名称。如果想获取列在表中的位置，可以使用元数据函数COLUMNPROPERTY(对象ID,列名,'ORDINAL')。

在sys.columns视图中，列的数据类型也是用ID表示的。若要返回该类型的名称，可和sys.types视图关联，或使用元数据函数TYPE_NAME。下面的SQL语句查询数据库中所有长度为10的字符串类型的列：

```
SELECT OBJECT_NAME(object_id) object_name,name
FROM sys.columns
WHERE TYPE_NAME(user_type_id) LIKE '%CHAR' AND
max_length=10
```

元数据函数与目录视图结合起来可使编写的SQL语句更为简洁。下面的SQL脚本用元数据函数检查t1表的c1列上创建了哪些主键、唯一约束或索引，代码量比前面的方式要少得多。

```
-- index name or primary key/unique constraint list
SELECT name index_name
FROM sys.index_columns IC
INNER JOIN sys.indexes I
ON IC.object_id=I.object_id AND IC.index_id=I.index_id
WHERE I.object_id=OBJECT_ID('t1') AND COL_NAME(I.
object_id,column_id)='c1'
```

输出SQL脚本

在熟悉了访问SQL Server目录结构的方法后，我们现在来看如何把获取的信息直接以SQL脚本的形式输出。程序员可以通过编程的方式拼装SQL语句，不过本文只讨论如何运用T-SQL来关联这些信息。大家后面会看到，相比普通的编程语言，SQL有着特有的优势。

CASE函数介绍

现假定一个数据库需要把各个表中所有原来定义为VARCHAR(10)的列更改为VARCHAR(20)，则可通过下面的SQL语句生成升级所需的脚本而不用逐列修改：

```
SELECT 'ALTER TABLE '+OBJECT_NAME(object_id)
+' ALTER COLUMN '+name+' VARCHAR(20)'
FROM sys.columns
WHERE object_id IN (SELECT object_id FROM sys.tables)
AND TYPE_NAME(user_type_id)='VARCHAR' AND max_length=10
```

为了防止表、列等的名称中有空格或与关键字冲突，名称部分最好用中括号括起来，形如'['+name+']'。或者使用QUOTENAME函数。为便于阅读，本文示例代码没有这样做。

上面的脚本有不完善的地方。列在修改定义时，如果不明确指明是否允许空值，则数据库按预先设置处理（通常是允许为空），而不是保留原来的设置。这可能带来不良后果，甚至生成的SQL语句根本就不能执行，如果相应的列上面建有约束的话。下面是针对是否允许为空的问题修改后的脚本。

```
SELECT 'ALTER TABLE ' + OBJECT_NAME(object_id)
+ ' ALTER COLUMN ' + name + ' VARCHAR(20) NULL'
FROM sys.columns
WHERE object_id IN (SELECT object_id FROM sys.tables)
AND TYPE_NAME(user_type_id)='VARCHAR' AND max_length=10
AND is_nullable=1
UNION ALL
SELECT 'ALTER TABLE ' + OBJECT_NAME(object_id)
+ ' ALTER COLUMN ' + name + ' VARCHAR(20) NOT NULL'
FROM sys.columns
WHERE object_id IN (SELECT object_id FROM sys.tables)
AND TYPE_NAME(user_type_id)='VARCHAR' AND max_length=10
AND is_nullable=0
```

代码的长度比原来增加了差不多一倍。如果再增加一个区分条件，则代码还会成倍增加。

T-SQL中有一个更好的解决方案，这就是CASE函数。说是函数，其实形式上更像一种运算。CASE函数有两种格式，一种类似于C/C#中的switch语句，如CASE season WHEN 1 THEN '春' WHEN 2 THEN '夏' WHEN 3 THEN '秋' ELSE '冬' END；另一种与C/C#中的三目运算符（？：）有些相似，如CASE WHEN @a>@b THEN @a ELSE @b END。我们把上面的脚本改用CASE函数的方式：

```
SELECT 'ALTER TABLE ' + OBJECT_NAME(object_id)
+ ' ALTER COLUMN ' + name + ' VARCHAR(20) ' +
CASE is_nullable
  WHEN 0 THEN 'NOT NULL'
  ELSE 'NULL'
END
FROM sys.columns
WHERE object_id IN (SELECT object_id FROM sys.tables)
AND TYPE_NAME(user_type_id)='VARCHAR' AND max_length=10
```

代码又减少到差不多原来的长度。需要判定的地方越多，CASE函数的优势越明显。

生成数据表定义脚本

生成数据表的定义脚本是生成T-SQL语句的基本功。下面我们从目录视图中还原一个表（不妨叫t1）的创建脚本。这里针对常见的场合，假定各个列的数据类型较为简单，要

么是整数(INT、SMALLINT、TINYINT),要么是字符串(CHAR、NCHAR、VARCHAR、NVARCHAR)。先不考虑约束、索引、标识列等细节，下面的SQL脚本输出表定义中最基本的部分。

```
SELECT 'CREATE TABLE t1 ('
UNION ALL
SELECT name+' '+TYPE_NAME(user_type_id)+
CASE
  WHEN TYPE_NAME(user_type_id) LIKE '%CHAR'
    THEN '('+
    CASE
      WHEN max_length=-1 THEN 'MAX'
      ELSE CAST(max_length AS VARCHAR(20))
    END +')'
  ELSE ''
END + ','
FROM sys.columns
WHERE object_id=OBJECT_ID('t1')
UNION ALL
SELECT ')'
```

下面我们看看如何生成约束定义脚本。这里我们仅关注其基本部分，即它包含哪些列。注意可能不止一列。如果我们已知约束或索引的名称，则生成定义脚本的过程有些类似生成表定义脚本。下面分别是生成索引定义脚本的SQL语句。

```
SELECT 'CREATE' +
CASE is_unique
  WHEN 0 THEN ''
  ELSE ' UNIQUE'
END +
CASE
  WHEN index_id=1 THEN ' CLUSTERED'
  ELSE ''
END + ' INDEX ' + name +' ON '+OBJECT_NAME(object_
id) +'('
FROM sys.indexes
WHERE name='索引名称'
UNION ALL
SELECT
CASE key_ordinal
  WHEN 1 THEN ''
  ELSE ','
END + COL_NAME(IC.object_id,column_id) +
CASE is_descending_key
  WHEN 0 THEN ''
  ELSE ' DESC'
END
FROM sys.index_columns IC
INNER JOIN sys.indexes I
ON IC.object_id=I.object_id AND IC.index_id=I.index_id
WHERE key_ordinal>0 AND name='索引名称'
UNION ALL
SELECT ')'
```

动态执行SQL脚本

在无人值守的场合，比如SQL Server作业，仅仅输出这些SQL脚本是不够的，还需要自动执行这些生成的SQL语句。SQL Server中用EXECUTE语句执行字符串形式的SQL脚本，被执行的字符串用括号括起来，这被称为SQL语句的动态执行。EXECUTE语句另一项众所周知的功能是执行存储过程，此时没有使用括号。

除了EXECUTE语句外，系统存储过程sp_executesql也可以动态执行SQL语句，不过必须是Unicode字符串。使用sp_executesql时可嵌入参数，包括输入参数和输出参数。

T-SQL的聚合赋值

现在考虑如何动态执行根据系统目录信息生成的SQL脚本。如果生成的脚本是单行的，这自然比较容易。但像上面输出多行信息的场合，得先把每行文本拼凑起来。简单地用字符串串联(+)代替UNION ALL的方式通常并不能奏效，因为UNION ALL前后的结果都可能是多行的记录，甚至行数也是未知的。我们如何把任意多行的字符串合并到一个字符串中呢？

大概大家都会想到通过游标读取每行字符串，逐一串联起来。这个办法管用，却不太理想。游标太过程化了，SQL非过程化的简洁性在使用游标时荡然无存。光是游标的创建、打开、关闭、销毁就得4条语句，还不包括WHILE循环。

如果我们可以创建一个自定义聚合函数，负责把各行的字符串连接起来就好了。

SELECT语句大家都很熟悉。SELECT关键字最基本的功能是从表格对象中查询数据，也可以跟SET关键字一样，为局部变量赋值。不过SELECT比SET强大得多，可以一次给多个变量赋值。甚至可以与FROM联手，把查询结果中的值直接赋值给变量，形如SELECT @local_variable =column_name FROM table_name，这通常用在查询结果只有单行的场合。如果查询结果有多行，则中间赋给变量的值会被后面的赋值覆盖，只有最后一次的赋值保留下来。如果SELECT 语句没有返回行，变量将保留当前值。

这其中隐藏的奥秘在于变量自己也可以被放在等号（=）的右面。假定这里有一个名为t1的表，看看下面的SQL脚本，您猜输出的值是多少？

```
DECLARE @local_variable INT
SELECT @local_variable = 0
SELECT @local_variable = @local_variable + 1 FROM t1
SELECT @local_variable
```

输出的结果不是1，而是t1中行的数量！上面的语句与SELECT COUNT(*) FROM t1是等效的，SUM、MAX、MIN等聚合函数也可以通过类似方法模拟。这种方式与CASE函数结合，可实现非常复杂的统计。

对于字符串串联，同样可以采用类似方式。执行下面的SQL脚本在单行中输出表t1的所有列名称，列之间用逗号分隔。

```
DECLARE @sql VARCHAR(MAX)
SELECT @sql = ''
SELECT @sql = @sql + name + ','
FROM sys.columns
WHERE object_id=OBJECT_ID('t1')
SELECT @sql
```

尽管联机手册上没有明确说明，但市场上见得到的所有SQL Server版本甚至SYBASE都支持这种方式的赋值。本文把这种现象称为变量的聚合赋值。需要指出，当这种方式的SQL语句中包含复杂的ORDER BY子句时，聚合赋值可能失效。

现在我们使用聚合赋值生成一个表的所有外键定义脚本。我们可以把根据名称生成单个外键定义脚本封装到标量函数中，再对每个外键逐一调用相应的标量函数即可。

```
CREATE FUNCTION GetForeignKeySQL(
      @constraint_name SYSNAME
)
RETURNS VARCHAR(MAX)
AS
BEGIN
  DECLARE @sql VARCHAR(MAX)
  SELECT @sql = 'ALTER TABLE ' + OBJECT_NAME(parent_
object_id)
+ ' ADD' +
  CASE is_system_named
    WHEN 1 THEN ''
    ELSE ' CONSTRAINT ' + name
  END + ' FOREIGN KEY('
  FROM sys.foreign_keys
  WHERE name=@constraint_name
  SELECT @sql = @sql +
  CASE constraint_column_id
    WHEN 1 THEN ''
    ELSE ','
  END + COL_NAME(parent_object_id,parent_column_id)
  FROM sys.foreign_key_columns
  WHERE constraint_object_id=OBJECT_ID(@constraint_name)
  SELECT @sql = @sql + ') REFERENCES '
+ OBJECT_NAME(referenced_object_id) + '('
  FROM sys.foreign_keys
  WHERE name=@constraint_name
  SELECT @sql = @sql +
  CASE constraint_column_id
    WHEN 1 THEN ''
    ELSE ','
  END + COL_NAME(referenced_object_id,referenced_
column_id)
  FROM sys.foreign_key_columns
  WHERE constraint_object_id=OBJECT_ID(@constraint_name)
  SELECT @sql = @sql+')'
  RETURN @sql
END
```

下面的SQL语句使用GetForeignKeySQL函数输出表t1所有的定义脚本。如果需要，也可以改为聚合赋值的方式输出单个字符串。

```
SELECT dbo.GetForeignKeySQL(name)
FROM sys.foreign_keys
WHERE parent_object_id=OBJECT_ID('t1')
```

快速清空数据表

现在我们来看如何利用动态执行和聚合赋值解决管理中遇到的棘手问题。这里以快速清空数据表为例。在SQL Server中，如果数据表中有大量记录的话，执行DELETE语句会死得很难看。有的读者会想到TRUNCATE TABLE语句，这的确是个好办法。不过使用TRUNCATE TABLE有限制。如果待删除的表被其他表外键约束引用的话，则即使其他表是空的，也不能使用TRUNCATE TABLE语句。这里的解决办法是先删除这些外键约束，待表清空后再原样恢复。删除外键前需先记录外键的定义脚本，这可借用前面的GetForeignKeySQL函数。下面是快速清空数据表的SQL脚本：

```
DECLARE @sql VARCHAR(MAX),@table_name SYSNAME
SELECT @sql = '',@table_name='替换为被删除的表名'
-- 删除引用该表的外键
SELECT @sql = @sql + '
ALTER TABLE ' + OBJECT_NAME(parent_object_id) + '
DROP CONSTRAINT ' + name
FROM sys.foreign_keys
WHERE referenced_object_id = OBJECT_ID(@table_name)
-- 清空数据表
SELECT @sql=@sql+'
TRUNCATE TABLE ' + @table_name
-- 按原样创建被删除的外键
SELECT @sql = @sql + '
' + dbo.GetForeignKeySQL(name)
FROM sys.foreign_keys
WHERE referenced_object_id = OBJECT_ID(@table_name)
EXEC(@sql)
```

结论

在SQL Server 2005/2008中可通过目录视图、元数据函数等方式访问数据库的元数据信息。利用T-SQL提供的聚合赋值和动态执行功能，结合CASE函数等各种字符串处理方法，可以在执行过程中动态生成简洁的SQL脚本，代替设计时硬编码方式的脚本编写，使相同的脚本可以不经修改应用到完全不同的场合。这类似JAVA或.NET中的反射机制。本文把这种方式称作Transact-SQL的元编程。

参考资料

1.《SQL Server2005联机丛书》，Microsoft

2.《SQL Server7.0技术内幕》，北京大学出版社，Ron Soukup & Delaney

3.《Inside Microsoft SQL Server 2005: T-SQL Querying》，Microsoft Press，Itzik Ben-Gan, Lubor Kollar, Dejan Sarka

4.《Inside Microsoft SQL Server 2005: T-SQL Programming》，Microsoft Press，Itzik Ben-Gan, Dejan Sarka, Roger Wolter

作者简介



王小鉴，拥有近10年Win32/.NET平台软件与数据库架构设计及研发经历，现为CSDN学生大本营数据库讲师（http://student.csdn.net/?106456）。目前关注改进软件质量的技术方法。欢迎通过电子邮件wang_xiao_jian@163.com与他交流。

责任编辑：郭晓刚 (guoxg@csdn.net)

在Spring下分布式事件通知服务框架的实现

在分布式系统中，分布在不同机器上的服务之间如何协同工作是一个非常关键的问题，事件通知则几乎是解决这个关键问题不二选择。

■ 文 / 刘理樵

前言

在分布式系统中，分布在不同机器上的服务之间如何协同工作是一个非常关键的问题，事件通知则几乎是解决这个关键问题不二选择，因此如何设计和搭建一个高效的，可扩展的事件通知服务就成为摆在很多系统分析人员和系统架构人员面前的一个问题。

因为篇幅的问题，对文章中涉及到的大部分代码都在QQ空间 http://654846813.qzone.qq.com/ 提供了演示。

总体规划

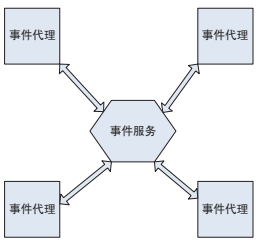


图1 事件通知服务

分布式事件通知服务的整体布局如图1所示。整个事件通知服务分成两大功能模块：事件服务和事件代理。事件服务是服务的核心，它承担着事件收集和事件转发的功能。事件代理之间不能直接通讯，必须要由事件服务提供中转。事件代理和事件服务之间的通讯是双向的。

我们把这个布局放置在整个系统下进行考虑，布局是如何实现不同服务之间的事件通讯的（图2）。

1．域内的通讯，如图2中箭头1所标识的。在这种情况下由事件代理行使事件服务模块的功能，使得事件在域内就可以直接通讯，而不要通过事件服务；

2．事件代理所在域和事件服务所在域的通讯，如图2中箭头2，4所标识的。在这种情况下，事件代理和事件服务分别完成事件发送和事件接收（箭头2所示）或事件接收和事件发送（箭头4所示）；

3．事件代理所在域之间的通讯，如图2中箭头3所示。因为事件代理之间不能直接通讯，所以事件服务承担了接收再转发的职责。

从对以上两个示图的分析和我们对事件通知的理解中，我们可以总体规划出在我们这个分布式事件通知服务框架设计中要涉及的内容：

1．在网络传输和功能模块操作中事件的表现方式；

2．事件的处理方式；

3．事件服务和事件代理的设计；

4．事件服务和事件代理之间通讯的设计。

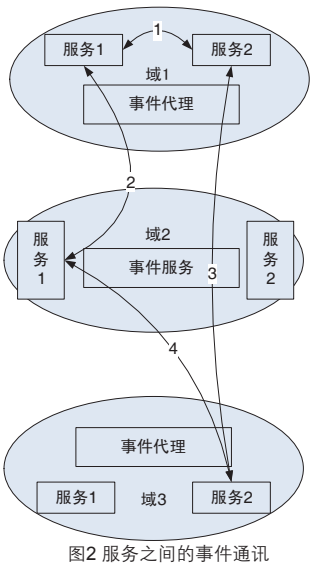


图2 服务之间的事件通讯

下面我们就针对这四个方面的内容由抽象到具体的逐一进行设计，并且在设计中引入Spring的框架，使得最后搭建起来的分布式事件通知服务框架更具灵活性和可配置性。

框架设计

1．事件的表现方式

事件是整个分布式事件通知框架中最为基本的元素，与事件相关的种种频繁的操作总结起来无外乎两种情况，第一种情况是事件以某种方式在网络介质中进行传输；第二种情况是事件以Bean的方式被各功能模块操作。而这两种方式之间要可以简单、快速地进行转换。

基于以上结论，我们在对事件的设计上采用Castor工具，来实现上述的对事件的要求。Castor是一种进行数据绑定的开源项目，通过使用Castor可以使得事件：易于扩展，Castor通过XML Schema自动生成Java Bean，因此只要修改配置文件

就可以完成事件相关信息的扩展；易于转换，Castor封装了事件两种表现方式之间转换的细节，使得转换变得很容易。

2．事件的处理方式

我们设计事件的处理方式为注册和广播方式：服务向事件通知服务注册它们感兴趣的事件，当事件发生时，事件通知服务向服务广播当前事件，服务侦听，如果该事件就是它注册过的事件，服务就调用相关业务逻辑进行处理。如图3所示。

3．事件服务和事件代理的设计

事件服务是整个事件通知服务的核心，它包括了两个部分：事件管理部分和事件处理部分。事件管理部分主要是提供API，这些API一部分供其他服务进行事件的注册，事件的发送之用，另一部分供事件服务广播事件之用；事件处理部分的工作包括了对事件信息的扩展，事件信息的入库。事件信息扩展主要是出于下面的考虑：有些服务是部署在不能进行数据库连接的服务器上，因此这些服务所能了解的系统的相关信息有限，比如它只知道源宿主机的IP地址，而不知道源宿主机在系统中的资产ID（这些信息保存在数据库中），因此事件服务在获得这类信息后要把相关的信息（比如资产ID）扩充到事件信息中，再把这个事件信息广播到其他服务中。

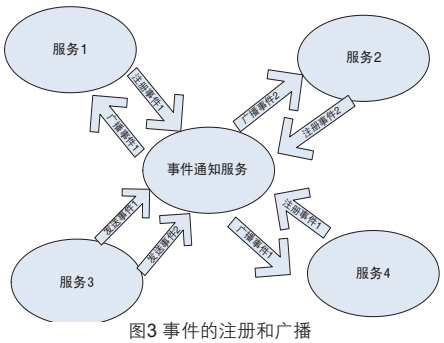


图3 事件的注册和广播

事件代理是简化版的事件服务，我们根据实际的情况对事件服务两部分功能进行删除，就形成了事件代理。因此我们引进了Spring框架，使得所有的修改都是通过配置完成。

4．事件服务和事件代理之间通讯的设计

事件服务和事件代理之间的通讯是分布式事件通知服务的基础，它们之间的通讯是双向的，如图4所示。

TcpProxy是寄存在本端主机上的对端主机的代理。事件代理所在域只会和事件服务所在域进行通讯，因此事件代理所在域只会包含一个事件服务TcpProxy。事件服务所在域要和所有事件代理所在域通讯，因此事件服务所在域会包括多个事件代理TcpProxy。

框架的具体实现

1．事件的表现方式

从前面的分析中，我们得出事件的两种不同的表现方式：XML和Java Bean，并且通过Castor作为这两种表现方式的转换工具。这里分为三个部分：1)通过Schema自动生成Java类；2)通过Castor的编组操作，完成从Java对象向XML的转换（见QQ空间《在Spring下分布式事件通知服务框架的搭建代码部分》代码1部分，简称见代码1，下同）；3）Castor的解组操作，完成XML向Java对象的转换（见代码2）。

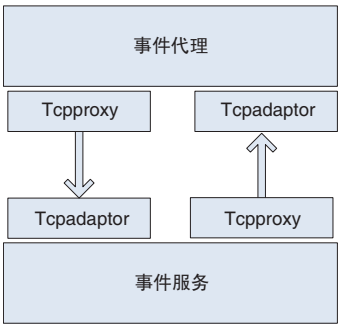


图4 事件服务和事件代理之间的通讯

2．事件的处理方式

从前面的分析中我们得出事件的处理分为四个部分：侦听、注册、发送、广播。

我们为每个需要侦听事件的服务创建一个事件监控线程，事件监控线程由一个事件队列和一个事件侦听器构成。事件队列是一个先入先出的队列，而事件侦听器其实就是一个接口，服务通过实现该接口来定义自己的事件处理逻辑。一旦有事件发生，事件服务或事件代理就会把该事件插入到相应的事件队列，事件侦听器就会取出列头的事件进行处理。

外部事件即为从别的域发送过来的事件，也就是图2中展示的箭头2，3，4。内部事件就是图2中的域内通讯。事件监控线程以统一的方式对外部事件和内部事件进行处理。

事件的注册也分为对内部事件的注册和对外部事件的注册。我们分别讨论。内部事件注册，就是服务通知事件服务或者事件代理，我对那个事件或那批事件感兴趣，前面已经提到了如果服务想侦听事件它就需要实现一个事件侦听接口（代码3）。外部事件注册就是特指事件代理所在域的服务，向事件服务注册自己感兴趣的事件。注册外部事件分两个过程：1）服务向服务代理注册内部事件；2）事件代理再向事件服务注册外部事件。外部事件注册的具体过程如图5所示，橙色箭头标出了三个步骤。

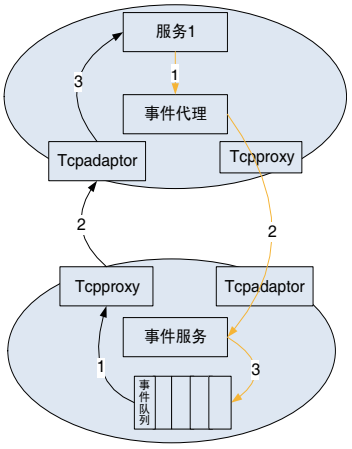


图5 事件的注册和广播

接下来是事件的发送。相对于其他事件处理逻辑，事件发送是最为简单的，它不分内部和外部，只有一种处理方式：

```
public synchronized void sendNow(Event event) {
  //把事件发送到事件服务或事件代理的处理队列中
  try {
    m_eventHandlerPool.getRunQueue().add(
      new EventHandler(event ,m_getNextEventIdStr));
  }
 ...
}
```

最后是事件广播。在事件发生时，事件代理或事件服务会通知相应的服务，这就包括了处在同一域内的服务，也有不在同域内的服务。

- 同域内事件广播。就是针对事件队列和两个map（事件名与事件队列组成的map，侦听器名和监控线程组成的map）进行操作（见代码6）；
- 异域事件广播，异域事件广播的示图就是图5的黑色箭头所示（在服务代理上，也是相同的逻辑）。在处理异域事件广播时要注意防止事件回环（具体做法见代码7）。

在前面的分析中，我们指出在处理异域事件广播时，事件服务和事件代理是使用相同的处理逻辑，其实两者是存在区别的：事件服务中的各个事件代理TcpProxy只会侦听自己注册过的事件；事件代理中的事件服务TcpProxy会把所有在宿主主机上发生的事件发送到事件服务上。为了解决这两者之间的区别，我们设计了一个特殊的侦听器，这个侦听器侦听所有的事件（见代码8）。一旦在事件代理上有事件发生，会首先调用这个特殊的侦听器，把事件发送给事件服务。

3．事件服务和事件代理的设计

事件服务和事件代理这两个模块都包含了两大功能部分：事件管理部分和事件处理部分。在这两模块的设计中，我们结合Spring框架，使得这两个模块的转换可以通过配置来进行。

事件管理部分主要的功能是提供API供其他服务进行事件注册和事件发送以及事件服务或事件代理进行事件广播时所用。图6就是事件管理部分的类图。

事件处理部分的主要功能是在事件发生时，事件服务和事件代理对事件进行的处理。在这部分功能上，事件服务和事件代理存在着差别：事件代理一般只会把事件广播出去，让其他服务的侦听器去处理；事件服务除此之外还要对事件进行相关附加信息的添加（从数据库中来），和进行事件的入库工作。事件处理部分的类图如图7。

我们引入Spring框架，使得事件处理部分功能点通过配置来完成。

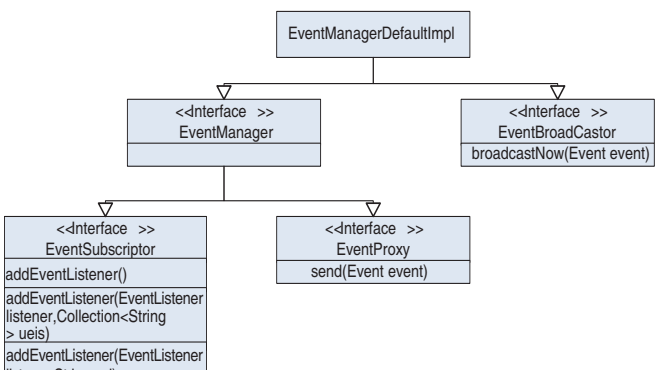


图6 事件管理部分的类图

```
<bean id="eventdEventHandler" class="org.transfar.
netmgt.eventd. EventHandlerDefaultImpl">
  <property name="eventProcessors">
    <list>
      <!--事件服务定义了三个事件处理功能：事件扩展，事件入库
和事件广播-->
      <ref local="eventExpander"/>
      <ref local="eventPersistent"/>
      <ref local="eventBroadcastProcessor"/>
    </list></property></bean>
  ...
<bean id="eventdEventHandler" class="org.transfar.
netmgt.eventd.EventHandlerDefaultImpl">
  <property name="eventProcessors">
    <list>
      <!--事件代理定义了一个事件处理功能：事件广播-->
      <ref local="eventBroadcastProcessor"/>
    </list></property></bean>
```

通过对配置文件的修改，我们日后可以很方便的对事件服务和事件代理的这部分功能进行扩展和修改。

4．事件服务和事件代理之间通讯

从图5中我们可以得出，事件服务和事件代理之间的通讯是由TcpProxy和TcpAdaptor完成的。TcpProxy专司发送，TcpAdaptor专司接收；事件代理只会向事件服务发送事件消息，因此事件代理上只会寄宿一个指向事件服务的TcpProxy；事件服务会发送事件消息到每个跟它相连的事件代理，因此它上面会寄宿多个TcpProxy，每个TcpProxy指向一个特定的事件代理；不论在事件代理还是事件服务上，TcpAdaptor都只有一个。

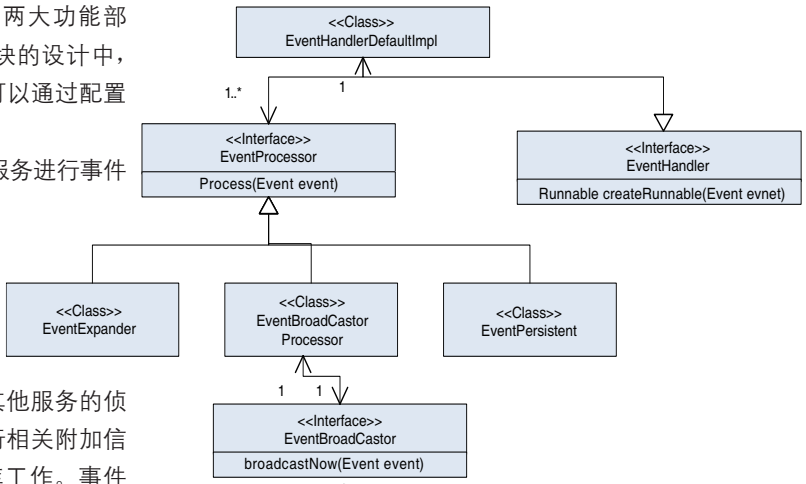


图7 事件处理部分的类图

同样我们通过Spring框架来配置TcpProxy。

```
<bean name="eventServiceProxy"  class="org.transfar.
netmgt.utils.TcpProxy">
  <!--通过查询eventdConfig.eventServerIP属性获得事件服
务的IP-->
  <property name="socketIPAddress">
    <util:property-path path="eventdConfig.
eventServerIP" />
```

（下转第107页）

.NET安全系列二：权限的运用

权限是CAS中的核心概念。如何正确的运用权限是影响.NET应用安全的关键。

■ 文 / 褚诚云

概述

在.NET安全系列上一篇文章里，我们介绍了.NET Framework的安全架构CAS。CAS中一个核心概念就是权限（permission）。权限确定代码是否可以执行对特定资源的操作，例如对特定注册表信息的访问权限等等。在这篇文章中，我们会对.NET中权限的运用进行探讨，包括如何申请权限，如何要求权限，以及如何定制权限评估的过程等等。

权限的申请（request permission）

在CAS的介绍里，我们指出，一个.NET组件最终被赋予什么权限是通过以下流程来确定的：CAS首先收集组件的相关信息，从中确定该组件所在的代码组；然后，从安全政策中找到该代码组所对应的权限表；权限表中所包含的权限，就是CLR最后赋予该组件的权限。也就是说，不是.NET组件本身申请哪个权限就会得到该权限的。CLR是权限赋予的最终确定者，而不是.NET组件本身。

那么为什么还需要.NET组件本身去申请权限？其主要目的是为了提高.NET组件运行的正确性和降低滥用权限的可能。我们在介绍完申请权限的方式后会对此做进一步阐述。

申请权限有三种方式：

- RequestMinimum：声明.NET组件本身运行时所必需的最小的权限集合。
- RequestOptional：声明.NET组件本身运行时所期望的，但不是必需的权限集合。
- RequestRefuse：声明.NET组件本身运行时所不希望获取的权限集合。

RequestMinimum

下面举个具体的代码例子。我们创建一个XBAP（XAML Browser Application）应用。

```
using System.Security;
using System.Security.Permissions;

namespace permissionTest {
  public partial class Page1 : Page {
    public Page1() {
      InitializeComponent();

      try {
        new FileIOPermission(
          PermissionState.Unrestricted).Demand();
        label1.Content = "IO permission is granted";
      }
      // Catch the security exception and
      // inform the user that the application was
      // not granted FileIOPermission.
      catch (SecurityException) {
        label1.Content = "IO permission is not
granted";

    }
  }
}
```

在IE的intranet zone启动这个测试程序，屏幕输出"IO permission is not granted"。这是预期的，因为运行于intranet zone的XBAP应用缺省不具备文件读写操作的权限。

那么，如果我们在这个.NET组件的assembly文件中加入以下声明：

```
[assembly: FileIOPermission(SecurityAction.
RequestMinimum, Unrestricted = true)]
```

重新编译后，在IE的intranet zone启动这个测试程序。这个XBAP程序不会在IE中启动，其原因是以下的异常：

```
Startup URI: permissionTest.xbap
Application Identity: file:permissionTest.
xbap#permissionTest.xbap, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=c75289c300f4989f,
```

```
processorArchitecture=msil/permissionTest.exe,
Version=1.0.0.0, Culture=neutral, PublicKeyToke
n=c75289c300f4989f, processorArchitecture=msil,
type=win32

System.IO.FileLoadException: Could not load file
or assembly 'permissionTest, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=null' or one of its
dependencies. Failed to grant minimum permission
requests. (Exception from HRESULT: 0x80131417)
File name: 'permissionTest, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=null' --->
System.Security.Policy.PolicyException: Required
permissions cannot be acquired.
   at System.Security.SecurityManager.ResolvePolicy(
... ...
```

也就是说，当RequestMinimum指定的权限不能满足的时候，CLR会拒绝加载.NET组件。这样带来的好处是在使用了RequestMinimum之后，.NET组件的代码编写中就可以假设运行时已经拥有RequestMinimum中声明的权限，从而降低代码的复杂性：例如不需要额外检查是否已经拥有某个权限等等。

RequestOptional

RequestOptional声明.NET组件本身运行时所期望的、但不是必需的权限集合。如果RequestOptional权限不能满足的话，.NET组件仍然可以被CLR加载。例如在上面.NET组件的assembly文件中加入以下声明：

```
[assembly: FileIOPermission(SecurityAction.
RequestOptional, Unrestricted = true)]
```

在IE的intranet zone启动这个测试程序，XBAP应用被加载，并输出“IO permission is not granted”。

看到这里，大家可能会问：RequestOptional到底有什么用处呢？似乎有没有使用都没什么关系。我们将RequestMinimum中的例子改动一下：

```
using System.Security;
using System.Security.Permissions;

namespace permissionTest {
  public partial class Page1 : Page {
    public Page1() {
      InitializeComponent();
      try {
        new FileDialogPermission(FileDialogPermissi
onAccess.Open).Demand();
        label1.Content = "FileDialogPermission Open
Access is granted";
      }
      catch (SecurityException) {
        label1.Content = "FileDialogPermission Open
Access is not granted";
      }
    }
  }
}
```

这里我们检查是否拥有Open Access的FileDialogPermission。没有使用RequestOptional的时候，我们在IE中启动这个组件，得到“FileDialogPermission Open Access is granted”。这是预期的，因为运行于intranet/internet zone的XBAP应用缺省都拥有该权限。

但是如果我们在上面.NET组件的assembly文件中加入以下声明：

```
[assembly: FileIOPermission(SecurityAction.
RequestOptional, Unrestricted = true)]
```

重新编译运作，我们得到的输出是：“FileDialogPermission Open Access is not granted”！为什么会如此？

这就是使用RequestOptional所带来到效果：CLR会自动拒绝赋予没有在RequestOptional中声明的权限，即便是这些权限在缺省情况下应该被赋予。也就是说，在上面的RequestOptional声明使用后，所有非FileIOPermission的权限都被拒绝，所以该.NET组件就没有FileDialogPermission的权限了。

为什么CLR会自动拒绝赋予没有在RequestOptional中声明的权限？这是因为CLR认为RequestMinimum和RequestOptional构成了一个.NET组件运行时所可能需要的全部权限。没有在其中声明的权限，就不要赋予给该.NET组件了。这样可以降低该组件由于拥有自身不需要的权限而被滥用的可能性。

RequestRefuse

RequestRefuse相对简单。.NET组件用RequestRefuse通知CLR不要赋予它某些特定的权限。使用的例子如：

```
[assembly: FileIOPermission(SecurityAction.
RequestRefuse, Unrestricted = true)]
```

Request权限的小结

说了这么多，大家也许会觉得经过RequestMinimum，RequestOptional和RequestRefuse使用后，最终一个.NET组件能拥有的权限不好确定。其实不然，可以用下面的简单公式：

$$PFinal = ((PMinimum \cup POptional) - PRefuse) \cap Pdefault$$

Pdefault是CLR赋予该.NET组件的缺省权限集合。

在缺省未使用Request语句的情况下：PMinimum是空集，POptional是全集（full trust），PRefuse是空集。

具体使用中，大家一定要注意RequestOptional所带来的拒绝权限的作用。在绝大多数情况下，使用了RequestOptional就不需要同时使用RequestRefuse。例如如果只在RequestOptional中声明FileIOPermission后，就没有必要特别再使用RequestRefuse去拒绝FileDialogPermission权限了。

需要强调的是：RequestMinimum，RequestOptional和RequestRefuse是在.NET 1.0就引入的权限声明模式。随着.NET平台的发展，.NET组件的权限控制越来越趋向于使用应用域（Application Domain）来实现。关于Application Domain的介绍，我们会在以后的文章中涉及。

权限的要求（demand permission）

在前面我们介绍Request权限的例子中，已经出现了Demand语句。它是权限要求的关键。原理很简单，Demand语句会指定CLR去检查一个特定的权限是否被满足。如果安全检查失败的话，CLR就会产生一个SecurityException。

Demand的使用语法很简单，如下：

```
FileIOPermission x = new FileIOPermission(Permissio
nState.Unrestricted);
x.Demand();
```

何时使用Demand语句？

如果你的.NET应用中只是调用.NET 所提供的各种库的话，你需要使用Demand语句的机会不大。这是因为，在.NET自身所带库的实现中，都已经加入了相应的Demand权限检查。

例如，如果你的.NET应用需要调用System.IO.Directory类去创建一个目录，你是否需要在调用Directory.CreateDirectory方法之前先用Demand语句去检查此时是否拥有对应的FileIOPermission权限？没有必要。在Directory类的实现中已经加入对所需FileIOPermission权限的检查，所以完全不需要再加入一个重复的权限检查。

使用Demand语句最常见的情况是定制的权限（Custom Permission）去保护定制的资源。例如，你需要提供一个.NET类给其他.NET组件使用。在你的.NET类中，你会对一个定制的资源进行操作，例如一个特定的网络资源。那么，你的.NET类的实现中，就需要对该定制的资源所对应的定制权限使用Demand，这样就可以确保该.NET类的调用者拥有这个定制权限。关于如何创建和使用定制的权限，我们也会在以后的.NET安全系列中涉及。

使用Demand语句的另一种情况是希望提早检查权限。例如，如果你的.NET类的实现中，需要先执行一段非常复杂费时的运算，然后将运算的结果存入一个数据文件。你可以在.NET类的初始化过程中就Demand相应的文件访问权限，而不是在使用.NET的FileIO类时才做权限检查。

Demand的内部实现：Stack Walk

在Demand的内部实现中，对当前调用堆栈的每一层调用者，都会进行对Demand指定权限的安全检查。

例如函数A调用函数B，函数B调用函数C，函数C中使用Demand检查相应权限，在Demand的实现过程中，首先在函数B检查，如果安全检查失败，则会产生Security Exception。否则，再在函数A中检查。以此向上，直到该.NET应用的入口函数。这个在堆栈上层层遍历的安全检查过程，我们称为Stack Walk。

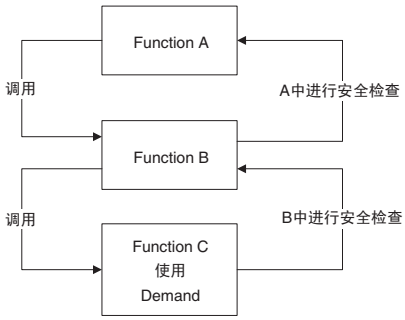


图1 .NET中Demand的Stack Walk

LinkDemand

LinkDemand的使用语法如下：

```
[CustomPermissionAttribute(SecurityAction.
LinkDemand)]
public static void foo() {
  ... ...
}
```

LinkDemand和Demand的区别是LinkDemand中不会进行一次完整的Stack Walk。它只会对上一层的直接调用者进行安全检查。在上面的例子里，如果Function C中使用LinkDemand，那么安全检查只会在B中进行，而不会在A中。

LinkDemand不进行完整的Stack Walk，所以它的使用有可能带来安全隐患。例如，一个未授权的函数A调用授权函数B来调用C，如果C中仅仅用LinkDemand保护的话，那么A就有可能在没有足够权限的情况下还可以对C中需要保护的资源进行访问。

我个人推荐不要使用LinkDemand方法，而是使用Demand。也许有的读者会觉得使用LinkDemand能带来性能上的提高。事实上，Stack Walk所花费的资源是非常少的。为了节省这么一点资源而引入可能的安全隐患非常不值得。

定制权限评估的过程

我们在上面介绍Stack Walk的过程中可以看出，安全权限的评估和检查的过程，就是Stack Walk的过程。那么，我们可以如何定制Stack Walk这个过程？.NET中提供了三种方法：Assert，Deny和PermitOnly。

Assert

Assert的使用语法如下：

```
FileIOPermission x = new FileIOPermission(Permissio
nState.Unrestricted);
x.Assert();
```

Assert的作用是定制CLR将Stack Walk对特定安全权限的检查停止在当前这一层。例如，在下面这个例子中，安全检查就会在函数C停止。对函数A和B的安全权限检查不会进行。

对Assert的使用者来说，使用Assert的.NET组件必须已经被赋予了需要Assert的权限，否则Stack Walk的安全检查失败，使用Assert也就没有意义。

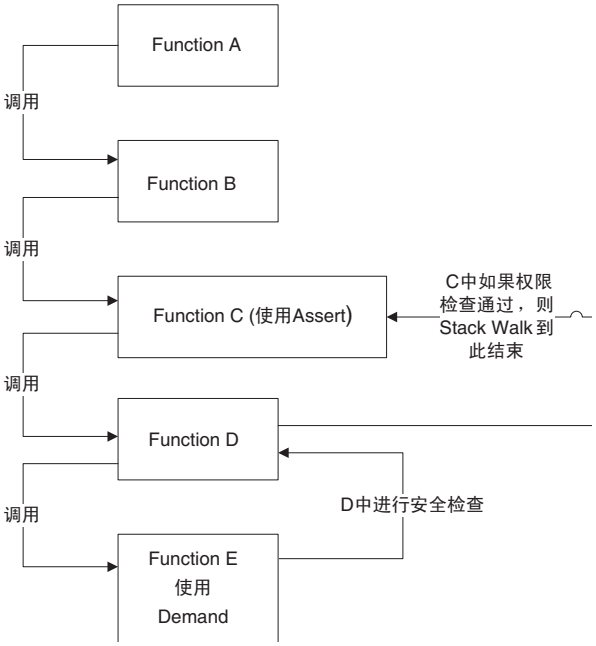


图2 .NET中Assert对Stack Walk的影响

Assert主要的使用方式是允许一些特定的调用者可以使用在通常意义下Stack Walk的安全检查会失败的调用。用图2举例：假设E中需要的是一个对特定文件的访问操作，C的设计中知道B或A所在的.NET组件是不拥有相应的文件访问权限的。C的设计者使用Assert作出保证：B或A是可信赖的。

通常情况下，在C中会通过其他的安全检查来确定调用者是可以信赖的B或A。否则，仅使用Assert的话可能会带来安全隐患。

Deny和PermitOnly

和Assert相反，Deny的作用是定制CLR将Stack Walk对特定安全权限的检查在当前这一层一定失败。

Deny的使用语法如下：

```
FileIOPermission x = new FileIOPermission(Permissio
nState.Unrestricted);
x.Deny();
```

同样用图2的这个例子：假设E中需要的是一个对特定文件的访问操作，C中对文件访问权限使用了Deny。那么，即使C已经被CLR赋予了文件的访问权限，Stack Walk的安全检查也会在C这一层失败。

PermitOnly和Deny的作用类似。区别在于Deny指明对某个权限的检查一定失败，而PermitOnly则指明只有对某个权限的检查才会成功。事实上的PermitOnly权限X的效果就是使用Deny在所有非X的权限上。

PermitOnly的使用语法如下：

```
FileIOPermission x = new FileIOPermission(Permissio
nState.Unrestricted);
x.PermitOnly ();
```

需要强调的是：不管是Deny还是PermitOnly，它们的用途都不是在安全意义上去限制外部调用者的权限。举个例子，在图2中，如果在函数D中存在Assert，则函数C中的Deny或是PermitOnly都可能被绕过。

Deny或PermitOnly主要目的是为了降低程序中可能出现滥用权限的错误。个人的推荐是仅仅使用它们在程序的测试过程中。如果出于安全方面的考虑需要对外部调用者进行隔离的话，推荐使用Application Domain。

总结

本文讨论了.NET的CAS安全架构下权限的运用方式，包括如何申请权限，如何要求权限，以及如何定制权限评估的过程等等。权限是CAS中的核心概念。如何正确的运用权限是影响.NET应用安全的关键。

（文章仅代表个人观点，与所在公司无任何联系。）

参考文献

1. Introduction to Microsoft .NET Security. IEEE Security & Privacy 6(6): 73-78 (2008). Chengyun Chu, Microsoft

2. .Net Security overview, http://msdn.microsoft.com/en-us/library/aa302422.aspx, Microsoft

3. Code Access Security in Practice, http://msdn.microsoft.com/en-us/library/aa302424.aspx, Microsoft

4. Writing Secure Code, Second Edition, Michael Howard, David LeBlanc, Microsoft

5. Assembly Level Declarative Security, .NET Security Blog, http://blogs.msdn.com/shawnfa/archive/2004/08/30/222918.aspx

作者简介



褚诚云，微软可信计算部（TWC）的安全软件开发工程师。2001年1月加入微软美国总部。先后申请多项数字版权保护及反病毒技术专利。近年来在微软技术大会上主讲过多个有关信息安全的技术讲座。中文Blog站点（http://blogs.itecn.net/blogs/chengyun_chu/）。

■ 责任编辑：郭晓刚 (guoxg@csdn.net)

OPhone联网应用开发中的线程管理与界面更新

使用AsyncTask处理联网，播放大尺寸媒体文件等较为耗时的工作，不但执行效率高，也可以节省代码。

■ 文 / 詹建飞

连接互联网的能力大大提升了移动电话的业务范围，中国移动很多出色的业务，像移动随身听、飞信等都是基于网络连接的应用程序，这也说明未来的移动互联网领域大有可为。因此深入掌握联网应用程序的开发和设计至关重要。本文主要介绍在开发联网应用过程中如何管理线程，如何从主线程之外更新界面。

为什么需要线程

假设需要开发一个联网应用程序，需要从一个网址抓取网页内容，这里读取的网页地址是笔者在本地机器上自己建立的服务器地址。当然在读取网页内容的时候，可以使用HttpClient提供的API，但是这并不是本文的介绍重点。缺乏联网程序开发经验的程序员可能写出下面的代码。

```
package com.ophone.network;

//这里为了节省篇幅，忽略了import项
public class NetworkActivity extends Activity {
  // 显示任务的执行状态和返回结果
  private TextView message;
  private Button open;
  private EditText url;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    message = (TextView) findViewById(R.id.message);
    url = (EditText) findViewById(R.id.url);
    open = (Button) findViewById(R.id.open);
    open.setOnClickListener(
      new View.OnClickListener() {
      public void onClick(View arg0) {
        connect();
      }
    });
  }

  private String connect() {
    try {
      HttpClient client = new DefaultHttpClient();
      // params[0]代表连接的url
      HttpGet get = new HttpGet(url.getText().toString());
      HttpResponse response = client.execute(get);
      HttpEntity entity = response.getEntity();
      long length = entity.getContentLength();
      InputStream is = entity.getContent();
      String s = null;

      if (is != null) {
        ByteArrayOutputStream baos =
          new ByteArrayOutputStream();
        byte[] buf = new byte[128];
        int ch = -1;
        int count = 0;
        while ((ch = is.read(buf)) != -1) {
          baos.write(buf, 0, ch);
          count += ch;
          // 为了在模拟器中清楚地看到进度，让线程休眠1000ms
          Thread.sleep(50000);
        }

        s = new String(baos.toByteArray());
      }
      // 返回结果
      return s;
    } catch (Exception e) {
      e.printStackTrace();
    }
    return null;
  }
}
```

网络连接通常是比较耗时的，尤其是在当前的GPRS这种低速率的网络情况下，这样connect()方法可能需要3~5秒，甚至更长的时间才能返回页面的内容。如果此连接动作直接在主线程，也就是UI线程中处理，会发生什么情况呢？为了在模拟器中更好的模拟网络读取速度慢的情况，笔者在读取过程中让线程休眠了50秒，运行NetworkActivity，点击“连接”按钮。意外发生了，按钮长时间没有反应，整个界面似乎是“死”掉了。系统随后显示出了ANR（应用程序无响应）错误提示，如图1所示。

在线程中联网

为什么出现ANR？答案是联网动作阻塞在了主线程，

长时间没有返回，这样OPhone弹出ANR错误。这个错误提示我们，如果某个任务可能需要长时间的运行才能返回，则必须把这个任务放置到单独线程中运行，避免阻塞UI线程。Java语言内置了对线程的支持，可以使用Thread类创建一个新线程，然后在run()方法中读取网页的内容，获得页面内容后调用TextView.setText()更新界面。修改后的connect()方法如下所示：



图1 阻塞主线程出现ANR错误

```
private void connect() {
  new Thread() {
    public void run() {
      try {
        HttpClient client = new DefaultHttpClient();
        // params[0]代表连接的url
        HttpGet get =
          new HttpGet(url.getText().toString());
        HttpResponse response = client.execute(get);
        HttpEntity entity = response.getEntity();
        long length = entity.getContentLength();
        InputStream is = entity.getContent();
        String s = null;

        if (is != null) {
          ByteArrayOutputStream baos =
            new ByteArrayOutputStream();
          byte[] buf = new byte[128];
          int ch = -1;
          int count = 0;
          while ((ch = is.read(buf)) != -1) {
            baos.write(buf, 0, ch);
            count += ch;
          }
          s = new String(baos.toByteArray());
          message.setText(s);
        }
      } catch (Exception e) {
        e.printStackTrace();
      }
    }
  }.start();
}
```

重新运行NetworkActivity，点击“连接”按钮。程序并没有像预期的那样获得网页的内容，并显示到TextView上。查看log可以看到在connect的执行过程中抛出了异常。接下来分析问题的所在。

使用Handler更新界面

其实，connect()方法中抛出的异常是由界面更新引起的。Connect()方法直接在新启动的线程中调用message.setText()方法是不正确的。OPhone平台只允许在主线程中调用相关View的方法来更新界面。如果返回结果在新线程中获得，那么必须借助Handler来更新界面。为此，在NetworkActivity中创建一个Handler对象，并在handleMessage()中更新UI。

```
//Task在另外的线程执行，不能直接在Task中更新UI，因此创建了
Handler
private Handler handler = new Handler() {
  @Override
  public void handleMessage(Message msg) {
    String m = (String) msg.obj;
    message.setText(m);
  }
};
```

当从connect()方法中获得网页内容后，使用如下方法更新界面。

```
s = new String(baos.toByteArray());
Message mg = Message.obtain();
mg.obj = s;
handler.sendMessage(mg);
```

重新运行NetworkActivity，点击“连接”按钮，结果如图2所示，正确读取了网页的内容。

AsyncTask

看上去修改后的connect()方法已经可用了，但是这种匿名线程的方式是存在缺陷的：第一，线程的开销较大，如果每个任务都要创建一个线程，那么应用程序的效率要低很多；第二，线程无法管理，匿名线程创建并启动后就不受程序的控制了，如果有很多个请求发送，那么就会启动非常多的线程，系统将不堪重负。另外，前面已经看到，在新线程中更新UI还必须要引入handler，这让代码看上去非常臃肿。

为了解决这一问题，OPhone在1.5版本引入了AsyncTask。AsyncTask的特点是任务在主线程之外运行，而回调方法是在主线程中执行，这就有效地避免了使用Handler带来的麻烦。阅读AsyncTask的源码可知，AsyncTask是使用java.util.concurrent框架来管理线程以及任务的执行的，concurrent框架是一个非常成熟，高效的框架，经过了严格的测试。这说明AsyncTask的设计很好地解决了匿名线程存在的问题。


图2 正确读取网页内容

AsyncTask是抽象类，子类必须实现抽象方法doInBackground(Params... p)，在此方法中实现任务的执行工作，比如连接网络获取数据等。通常还应该实现onPostExecute(Result r)方

法，因为应用程序关心的结果在此方法中返回。需要注意的是AsyncTask一定要在主线程中创建实例。AsyncTask定义了3种泛型类型Params，Progress和Result。

- Params 启动任务执行的输入参数，比如HTTP请求的URL。
- Progress 后台任务执行的百分比。
- Result 后台执行任务最终返回的结果，比如String。

AsyncTask的执行分为四个步骤，与前面定义的TaskListener类似。每一步都对应一个回调方法，需要注意的是这些方法不应该由应用程序调用，开发者需要做的就是实现这些方法。在任务的执行过程中，这些方法被自动调用。

- onPreExecute() 当任务执行之前开始调用此方法，可以在这里显示进度对话框。
- doInBackground(Params...) 此方法在后台线程执行，完成任务的主要工作，通常需要较长的时间。在执行过程中可以调用publicProgress(Progress...)来更新任务的进度。
- onProgressUpdate(Progress...) 此方法在主线程执行，用于显示任务执行的进度。
- onPostExecute(Result) 此方法在主线程执行，任务执行的结果作为此方法的参数返回。

PageTask扩展了AsyncTask，在doInBackground()方法中读取网页内容。PageTask的源代码如下所示：

```
/ 设置三种类型参数分别为String,Integer,String
class PageTask extends AsyncTask<String, Integer, String>{

  // 可变长的输入参数，与AsyncTask.exucute()对应
  @Override
  protected String doInBackground(String... params) {
    try {
      HttpClient client = new DefaultHttpClient();
      // params[0]代表连接的url
      HttpGet get = new HttpGet(params[0]);
      HttpResponse response = client.execute(get);
      HttpEntity entity = response.getEntity();
      long length = entity.getContentLength();
      InputStream is = entity.getContent();
      String s = null;

      if (is != null) {
        ByteArrayOutputStream baos =
          new ByteArrayOutputStream();
        byte[] buf = new byte[128];
        int ch = -1;
        int count = 0;
        while ((ch = is.read(buf)) != -1) {
          baos.write(buf, 0, ch);
          count += ch;
          if (length > 0) {
            // 如果知道响应的长度，
            // 调用publishProgress () 更新进度
            publishProgress((int) ((count / (float)
length) * 100));
          }
          // 为了在模拟器中清楚地看到进度，让线程休眠100ms
          Thread.sleep(100);
        }
        s = new String(baos.toByteArray());
      }
      // 返回结果
      return s;
    } catch (Exception e) {
      e.printStackTrace();
    }
    return null;
  }

  @Override
  protected void onCancelled() {
  super.onCancelled();
  }

  @Override
  protected void onPostExecute(String result) {
    // 返回HTML页面的内容
    message.setText(result);
  }

  @Override
  protected void onPreExecute() {
    // 任务启动，可以在这里显示一个对话框，这里简单处理
    message.setText(R.string.task_started);
  }

  @Override
  protected void onProgressUpdate(Integer... values) {
    // 更新进度
    message.setText(values[0]);
  }
}
```

执行PageTask非常简单，只需要调用如下代码。重新运行NetworkActivity，不但可以抓取网页的内容，还可以实时更新读取的进度。读者尝试读取一个较大的网页，看看百分比的更新情况。

```
PageTask task = new PageTask();
task.execute(url.getText().toString());
```

总结

本文介绍了OPhone联网应用开发中应该注意的两个问题：线程管理和界面更新。不但分析了问题的所在，也给出了多种解决方案。这里笔者推荐使用AsyncTask处理联网、播放大尺寸媒体文件等较为耗时的工作，不但执行效率高，也可以节省代码。

作者简介



詹建飞，毕业于北京邮电大学信息工程学院，硕士。喜欢读书，打篮球。是移动开发技术的狂热爱好者，熟悉Java ME，著有《J2ME开发精解》和《Java ME核心技术与最佳实践》，电子工业出版社2007年优秀作者。目前正在努力学习Android开发。

责任编辑：郭晓刚 (guoxg@csdn.net)



主持人：张银奎
《软件调试》一书作者，从事软件开发和研究十余年，对IA-32架构、操作系统内核、虚拟技术，尤其对软件调试有较深入研究。翻译（合译）作品包括《数据挖掘原理》、《机器学习》、《人工智能：复杂问题求解的结构和策略》等。

在调试器中品味计算机系统的睡眠过程（上）

大家都知道，当暂时不用计算机的时候，可以让系统进入低能耗的睡眠状态，等使用时再将其唤醒，恢复到睡眠前的状态，这样既方便快捷，又可以节约能源。但大家是否思考过，计算机是如何进入睡眠的呢？为什么有时候无法进入睡眠呢？事实上，进入睡眠是一个很复杂的过程，需要系统中很多部件通力协作，才能进入“美丽的梦乡”。

这一期，我们就以运行Windows操作系统的IA（英特尔架构）平台为例，与大家一起品味计算机系统进入睡眠的过程，我们先从一些基础知识开始。

睡眠的种类

与人类的睡眠有深睡眠和浅睡眠之分类似，计算机系统的睡眠也有深浅之分。今天的大多数个人计算机（PC）系统和服务器系统都是遵循ACPI（Advanced Configuration and Power Interface）规约设计的。简单说来，ACPI规约定义了四种睡眠状态，由浅到深依次为S1、S2、S3和S4。四种睡眠状态都属于G1这种全局状态，类似的还有，G0和S0代表工作状态，G2和S5代表软关机状态，G3代表机械方式停止供电的状态。图1画出了各个状态和状态之间的转换关系。

下面简要介绍一下S1~S4这四种睡眠状态：

- S1：所有CPU停止执行指令，但是CPU仍然处于供电状态，内存和其他主要部件也都处于供电状态，因此S1又被称为Power On Standby（供电待机），简称POS。
- S2：不再为CPU供电，内存处于刷新状态，维持现有数据，唤醒时，CPU从复位向量（Reset Vector）处开始执行，与重启类似。
- S3：CPU和系统中大多数部件处于关闭（off）状态，不再耗电，但内存仍然工作，保存着系统状态，唤醒时CPU从复位向量开始执行，根据保存在内存中的数据恢复状态。因此S3又称为挂起到内存（Suspend To RAM），简称STR。
- S4：所有状态信息保存到磁盘这样的永久介质中，包括内存在内的几乎所有系统部件都处于关闭状态；又称为挂起到磁盘（Suspend To Disk）或者休眠（Hibernate）。

在以上四个状态中，省电的效果依次递增，进入和退出相应状态所需的时间和复杂度也是递增的。S1、S2和S3有时被统称为待机（Standby）状态。

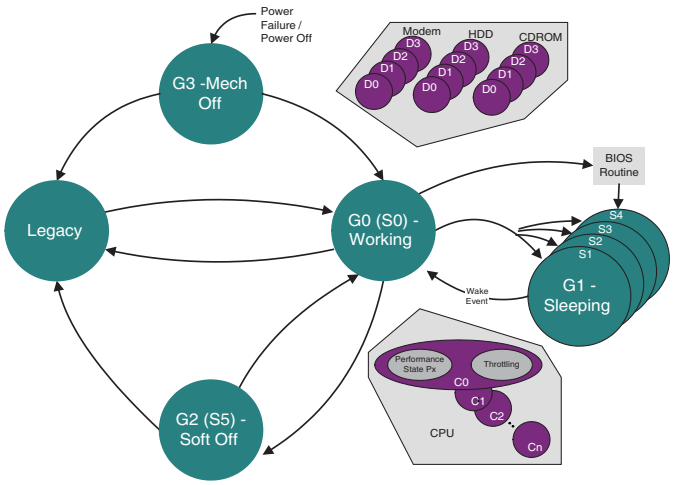


图1 ACPI定义的电源状态（摘自ACPI规约3.1b）

硬件平台所支持的睡眠状态

尽管ACPI规约定义了四种睡眠状态，但是并不要求每个平台都要支持全部四种状态。通常，硬件平台会根据实际情况实现其中的一种或多种，然后通过ACPI接口报告给操作系统。

操作系统支持的睡眠方式

考虑到S1、S3这样的名字不容易让普通用户理解，所以操作系统通常使用更直白的词语来描述睡眠状态。以Windows XP为例，它定义了待机（Standby）和休眠（Hibernate）两种睡眠状态，待机状态对应于S1到S3中的一种，休眠对应于S4。

当然，操作系统的睡眠方式是依赖于硬件平台的，如果平台缺少支持，那么相应的睡眠方式便会被禁止。举例来说，运行在Virtual PC虚拟机中的Windows XP就不支持待机和休眠，原因是虚拟机中的虚拟BIOS没有定义任何睡眠状态。

观察虚拟机平台的ACPI脚本（图2），可以看到它只定义了S0和S5两种状态。

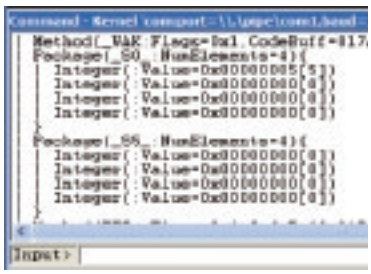


图2 在WinDBG中通过AMLI观察ACPI对象

图2显示的是通过内核调试会话与ACPI脚本解释器（ACPI Machine Language Interpreter，简称AMLI）调试引擎对话的场景。AMLI调试引擎只包含在检查版本（Checked）的ACPI驱动程序中，调试时需要先替换这个驱动，不熟悉的读者，请阅读以下链接所指示的文章：http://tinyurl.com/acpidbg。

因为进入待机状态需要设备驱动程序的配合，所以当系统中存在老式的设备驱动程序（legacy driver）时，Windows也会禁止待机功能，这可以解释为什么没有安装好驱动程序时系统无法进入待机状态。

下面我们将分步骤讨论系统进入S3状态的详细过程，操作系统以Windows 7（简称Win7）为主，兼顾Windows XP和Windows Vista。

第1步：发起睡眠请求

可以使用以下两个API之一来发起睡眠请求：

- BOOL WINAPI SetSystemPowerState(BOOL fSuspend, BOOL fForce);
- BOOLEAN WINAPI SetSuspendState(BOOLEAN Hibernate,BOOLEAN ForceCritical,BOOLEAN DisableWakeEvent);

这两个API的作用和参数都类似，调用方法略有差异，前一个是从Kernel32.dll导出的，后一个是从PowrProf.dll导出的；前一个需要当前线程具有SE_SHUTDOWN_NAME权限，后一个不需要调用者设置权限，SetSuspendState函数内部会调用SetPrivilegeAttribute自动设置SE_SHUTDOWN_NAME权限。

如果没有SE_SHUTDOWN_NAME权限就调用SetSystemPowerState()，那么它会返回FALSE，取LastError会得到0x522，含义是没有所需权限：

```
lkd> !error 0x522
Error code: (Win32) 0x522 (1314) - A required
privilege is not held by the client.
```

两个函数的工作原理也是一样的，都是调用内核服务NtInitiatePowerAction来发起睡眠动作，在ntdll!NtInitiatePowerAction处设置断点，然后观察栈回溯，可以清楚地看出这一点。

```
0:000> knL
 # ChildEBP RetAddr
00 0012fc38 74ad3896 ntdll!NtInitiatePowerAction
01 0012fc64 00401fd7 PowrProf!SetSuspendState+0x5b
02 0012fdb4 7e418734 PowerBoy!WndProc+0x427
0:000> knL
 # ChildEBP RetAddr
00 0012fc50 7c861b42 ntdll!NtInitiatePowerAction
01 0012fc68 0040205f kernel32!SetSystemPowerState+0
x2f
02 0012fdb4 7e418734 PowerBoy!WndProc+0x4af
```

另外值得说明的是，两个API都是以同步（阻塞）方式工作的，如果请求成功，那么要等系统进入睡眠，再唤醒后它们才返回。换句话来说，要等系统睡下去，再醒来后才返回。

图3显示了一个用于发起睡眠请求的小工具，PowerBoy，的工作场景。窗口的工作区是PowerBoy打印出的日志信息，第5行显示它在10:39分时调用SetSystemPowerState API发起睡眠请求，第9行显示10:41分这个调用才返回，期间系统进入睡眠，睡了一小会儿后被笔者唤醒了。

概括一下，NtInitiatePowerAction是内核中开始睡眠过程的起点，不论是SetSuspendState还是SetSystemPowerState，内部都会调用这个内核服务。

进入内核后，NtInitiatePowerAction在确认当前线程具有SE_SHUTDOWN_NAME权限后，会以独占方式获取全局的电源策略锁（PopPolicyLock），以确保不会有多个线程同时执行睡眠过程，接下来，它会调用PopSetPowerAction。

PopSetPowerAction内部会将要采取的动作设置到一个名为PopAction的全局数据结构中，并更新这个结构中的有关字段，待准备工作做好后，调用PopCompleteAction来设置一个事件对象，通知系统的工作线程来处理这个任务，于是PopPolicyWorkerThread（简称PPWT），会开始工作。PPWT内部会调用PopIssueActionRequest函数来处理放在PopAction结构中的任务。

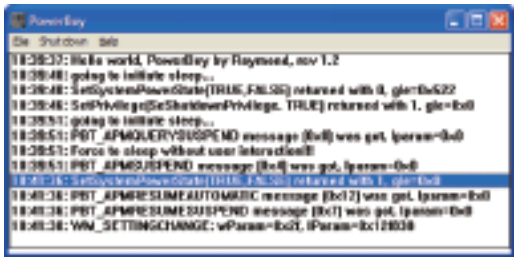


图3 用于发起睡眠请求的小工具

PopIssueActionRequest是睡眠和唤醒过程的“总导演”，接下来的睡眠动作和唤醒以后的恢复动作都是从这里发起的，它的前半部分是睡眠过程，后半部分是唤醒过程。如果是Vista或者Win7，而且启用了调试输出，那么它会先打印出如下一条消息：PowerAction: Sleep, Min=Sleeping3, Flags a0000002

然后开始依次采取下面要介绍的规定动作，首先是让使用所谓的Callout（“外调”）机制通过窗口子系统处理上层任务。

第2步：“外调”窗口子系统

图形显示（GDI）、用户输入和窗口都是由Windows子系统（窗口子系统）负责的，在进入睡眠状态前，需要把控制权交给窗口子系统，让它做各种准备工作，包括关闭显示、通知应用程序等。窗口子系统是按会话来组织的，每个会话有自己的窗口子系统服务器（CSRSS）、窗口站和桌面等。电源管理器（PopIssueActionRequest）如何把睡眠任务“踢出去”给Windows子系统呢？电源管理器的PopStateCalloutDispatch函数和PopStateCallout函数指针便是用于解决这个问题的。PopStateCalloutDispatch会遍历系统中的所有会话（Session），然后附加到每个会话，在这个会话中，调用PopStateCallout所指向的函数，通常是Windows子系统的内核态模块Win32K.SYS的UserPowerStateCallout函数。

```
kd> ln poi(nt!PopStateCallout)
Exact matches:
win32k!UserPowerStateCallout = <no type information>
```

UserPowerStateCallout会更新一个名为gPowerState的全局数据结构，然后调用QueuePowerRequest函数，向Win32k的电源请求队列（gPowerRequestList）中增加一项。接下来，如果当前线程是窗口线程，那么QueuePowerRequest会调用xxxUserPowerCalloutWorker来处理在当前线程中列表中的请求。

如果当前线程不是窗口线程（是系统线程或者其他非GUI线程），那么QueuePowerRequest会设置一个全局事件对象（gpEventPowerRequest），让当前会话的Windows子系统服务器进程（CSRSS）来处理这个请求。CSRSS中专门有一个工作线程，用于监视包括这个电源事件在内的多种事件，在收到这个事件后，这个线程会通过NtUserCallNoParam调用Win32K的内核服务，让Win32K来处理电源请求，这会导致xxxUserPowerCalloutWorker在CSRSS的工作线程中执行。xxxUserPowerCalloutWorker会从队列中取出请求，然后分发给xxxUserPowerStateCalloutWorker来处理。

值得说明的是，还有一组包含Event字样的函数和指针：nt!PopEventCalloutDispatch，nt!PopEventCallout，win32k!UserPowerEventCallout，win32k!xxxUserPowerEventCalloutWorker。它们是以事件方式工作的，每个事件是以一个事件ID来标识的。上面介绍的State字样的函数和函数指针是专门用于处理电源状态变化的，它们之间是以全局变量来沟通信息的。

第2A步：关闭显示器（Vista开始）

显示器是系统中消耗电能的主要部件，也是用户最容易注意到的部件，因此，从Vista开始，系统开始睡眠过程后，优先要做的一个动作就是关闭显示器，这样不仅可以快速得到省电效果，而且可以让用户觉得进入睡眠的速度非常快，尽管事实上，睡眠的工作才刚开始，后面“摸黑”要做的工作还有很多。

具体说来，Win32k的UserPowerStateCallout会调用QueuePowerRequest向电源请求队列中加于一个参数为空的请求，如上文所说，CSRSS的工作线程会处理这个请求，执行Win32k的xxxUserPowerStateCalloutWorker，后者会调用PowerOffMonitor函数来关闭显示器，下面的栈回溯显示了以上函数间的调用关系。

```
kd> kn
 # ChildEBP RetAddr
00 807aac6c 9114264c win32k!PowerOffMonitor
01 807aad00 910bd01c win32k!xxxUserPowerStateCallou
tWorker+0x7b
02 807aad18 911229e4 win32k!xxxUserPowerCalloutWork
er+0x34
03 807aad28 828723fa win32k!NtUserCallNoParam+0x1b
04 807aad28 77018244 nt!KiFastCallEntry+0x12a
```

第2B步：询问应用程序（Vista之前）

尽管运行在用户态的应用程序处于低特权的被管理地位，但它们是系统的载体，好比车上的乘客，没有它们系统也就没有了意义。基于这样的考虑，系统进入睡眠这样的重大事件应该征求应用程序的意见，询问它们是否同意进入睡眠状态。这便是WM_POWERBROADCAST（0x218）. PBT_APMQUERYSUSPEND（0）消息的设计初衷。

图4所示的栈回溯便是Explorer进程发起睡眠请求后，UserPowerStateCallout调用QueuePowerRequest增加电源请求，然后在同一个线程中（因为当前线程是GUI线程）执行xxxUserPowerStateCalloutWorker处理请求的过程。最上

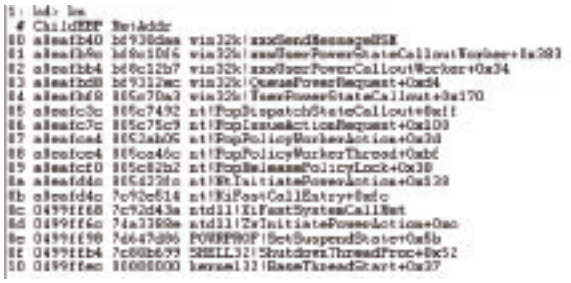


图4 发送询问睡眠消息的栈回溯（WinXP）

面的0号栈帧显示，xxxUserPowerStateCalloutWorker函数正在调用xxxSendMessageBSM来广播PBT_APMQUERYSUSPEND消息。xxxSendMessageBSM会遍历所有窗口站和桌面，然后向所有顶层窗口发送消息。

应用程序收到消息后，可以同意或者否定这次睡眠（详见MSDN中关于PBT_APMQUERYSUSPEND消息的说明）。但是从Vista开始，为了加快睡眠速度和防止糟糕的应用程序意外阻止睡眠，这个消息取消了。

第2C步：通知应用程序

与发送PBT_APMQUERYSUSPEND消息的过程类似，PopIssueActionRequest会使用上面说的Callout机制让窗口子系统使用xxxSendMessageBSM向所有顶层窗口发送WM_POWERBROADCAST(0x218)，PBT_APMSUSPEND(4)消息，通知应用程序系统将进入睡眠状态。因为最终是使用xxxSendMessageTimeout函数来发送消息的，所以即使应用程序不回复消息，那么超过执行的Timeout时间后，系统也会继续通知下一个程序。Timeout的值在Vista之前是20秒，Vista开始缩短为2秒。这意味着每个应用程序在收到PBT_APMSUSPEND消息时，还至少有2秒（或20秒）的时间来做睡眠前的准备工作。

第2D步：通知系统服务

在通知应用程序后，PopIssueActionRequest后使用“外调”机制来通知系统服务，不再赘述。

本期问题：

搜索一下图书和网络，很少有关于计算机的睡眠过程的详细介绍，是什么原因呢？本文尝试着去揭开这个过程的神秘面纱，抛砖引玉，欢迎您评点或者针对没有说清的地方提问？

编者说明：

- 投稿信箱：contest@csdn.net
- 解答提交时间，最好能早于当月15日。
- 联系方式写在一个单独的TXT文件里，包括以下几项：

1）姓名 2）工作单位或学校 3）电话联系方式
4）邮寄地址 5）E-mail地址

■ 责任编辑：郭晓刚（guoxg@csdn.net）

（上接第96页）

```
    </property>
      <!--通过查询eventdConfig.eventServerPort属性获得事件
服务的端口-->
    <property name="socketPort">
      <util:property-path path="eventdConfig.
eventSeverPort" />
    </property></bean>
...
<bean id="eventManagerImpl"  class="org.transfar.
netmgt.eventd.EventManagerDefaultImpl">
    <property name="handlerPoolSize" ref="eventManage
rHandlerPoolSize"/>
    <property name="eventHandler"
ref="eventdEventHandler"/>
    <property name="eventManagerProxy"
ref="eventManager"/>
    <!--在事件代理上配置事件服务tcpproxy-->
    <property name="eventServiceProxy"
ref="eventServiceProxy"/>
</bean>
...
<!—事件服务的配置中就没有eventServiceProxy属性的配置-->
<!—使得eventManagerImpl对象中的eventServiceProxy属性为
null-->
<!—事件服务使用DelegateProxyFactory 中的静态方法进行事件
代理<!--TcpProxy的管理-->
<bean id="eventManagerImpl"  class="org.transfar.
netmgt.eventd.EventManagerDefaultImpl">
    <property name="handlerPoolSize" ref="eventManage
rHandlerPoolSize"/>
    <property name="eventHandler"
ref="eventdEventHandler"/>
    <property name="eventManagerProxy" ref="eventManager"/>
</bean>
```

TcpProxy和上面介绍的EventManagerDefaultImpl都实现了接口EventPorxy中的Send方法，但是两者的业务逻辑是不同的，EventManagerDefaultImpl中send方法的说明上面已经谈过了，TcpProxy的send是通过网络把消息发送到对端主机上去（见代码9）。TcpAdaptor专司从网络上接收事件信息，并提交给宿主主机的事件服务或事件代理去处理（具体实现见代码10）。

结束语

到此我们对如何构建一个分布式的事件通知框架做了一个较为全面的分析，但是若要去构建一个兼顾效率、稳定、可扩展的事件通知框架，我们还有很多事情要去做。我希望这篇文章可以起到一个抛砖引玉的效果，为大家今后的系统搭建起到一个借鉴的作用！

作者简介



刘理樵，湖南创发科技有限公司程序员，从事Java开发和系统分析多年。如有兴趣在上述两个方面与之进行探讨，可以email联系，邮箱为llq@transfar.com。

■ 责任编辑：郭晓刚（guoxg@csdn.net）

本文节选自电子工业出版社博文视点最新图书《我是一只IT小小鸟》。一群IT小小鸟，在失望中追求希望，在迷茫中辨别方向，他们用自己的成长故事，致所有平凡而又不平常的人们：青春太宝贵，千万别浪费；要想不浪费，万事早准备。

半路出家，从化验员到程序员

■ 文 / 黄小明

引子：赶上了毕业生不再包分配的第一年

我是1994年考上大学的。那时高考填志愿，根本没考虑“我喜欢什么专业、我适合学什么专业、我以后的职业是什么”。尤其在农村，只要考上大学就行，于是我稀里糊涂被录取到兰州大学化学系。这一年，也是高校并轨招生改革的第一年，就是说，毕业生以后不再包分配，将实行双向选择。我刚刚考上，就在考虑这个问题，四年以后我找什么样的工作？我怎么准备？我家来自农村，没有任何亲人能帮忙，一切只能靠自己。

本科四年

每天看报纸，收集和分析就业信息

那个时候还没有网络，报纸是最主要的消息来源。我有一个习惯，就是喜欢看报纸。每天中午，从食堂买好饭菜，就在学校的阅报栏旁边，把《中国青年报》、《人民日报》、《参考消息》、《光明日报》、《文汇报》这些主流报纸都看完。在读这些报纸时，我经常关注：什么样的毕业生好找工作。报纸看多了，慢慢发现，要找好工作，除了学习专业，更重要的是锻炼综合素质，提高社会实践能力，加入学生会、入党，多拿几个证书。

当记者，入社团，搞零售——积累社会实践经验

大一下半学期，我加入了校广播电台当记者，相当长时间里都是发新闻最多的学生记者之一。同时也加入了系学生会，担任系刊编辑部的主编。除了在校内多参加活动，我还走出校门，从图书市场批发杂志去学校零售，周末帮日化公司上门促销，暑假帮广告公司拉业务。这些事，在如今北京、上海的高校看来都是司空见惯了，但当时在兰州这个相对闭塞的地方，还是很少见的。

由于参加了这么多社会实践，本科毕业时，我各方面已经有很大提升，辅导员说我是全系变化最大的同学。由于数学成绩一直最好，我最喜欢数学，并不喜欢化学这个专业，而且看报纸上说电子和计算机专业好找工作，数学基础好适合学计算机，这让我萌生了改专业学计算机的念头。但是当时想转系太难了，所以我一直还在化学系。

放弃本校保研，奔赴经济发达的广东

大四结束时，学校公布了保研名单，我刚好在最后一名。但就在从导师家回来的路上，我又在思考这个问题：我不喜欢化学，还要读这个化学硕士吗？如果我选择了读这个不喜欢的硕士专业，或许将来就只能在这个不喜欢的领域发展了。报纸上不是经常说广东的改革开放取得很大成就，非常适合想出去闯一闯的青年人。我应该先去广东闯一闯，等想清楚自己适合做什么再考研。就这样我放弃了保研，直接去广东找工作了。

工作七年

1998年，我大学毕业，来到了广东。虽说广东是改革开放的前沿，发展机会很多，但是主要是电子、计算机、通信方面的公司多。化学专业的毕业生想找一个合适的工作也不容易。三下广东，我去过广州、深圳、珠海、佛山、顺德，还逃过车票，最后被广东珠海丽珠制药厂接收了。

在很多同学眼里，珠海是经济特区，丽珠是名牌企业，大家都羡慕我的工作。实际上，我在化验室里做一名化验员，整天围着试管、PH试纸、酒精转，由于一直不喜欢化学这个专业，所以也在不断思考自己将来要做什么。

化验室里我最懂电脑

幸运的是，化验室里有一台电脑用于打印化验报告。有一次，有个同事用电脑遇到一点问题，我刚好在旁边，他问我“你是刚来的大学生，电脑肯定学得很好吧”。我跑过去拿鼠标点一个按钮，点好几次都点不中。虽然我在学校也接触过286和DOS操作系统，但是却没有用过Windows，也没有用过鼠标。这真让我觉得很没有面子：一个大学生居然连电脑鼠标都用不熟练。

那个时候下班也没有什么事情，我就去书店买了Windows、Word、Excel方面的书，在化验室的电脑上学习，很快成为化验室里最熟悉电脑的人。

努力自学编程，一心想转行

工作之后我依旧喜欢看报纸。报纸上都有很多招聘程序员的信息，有的只要求掌握C语言程序设计。这个

时候我想学习C语言然后去应聘，于是把这个想法告诉了一个也很熟悉电脑的同事。但他说我现在想学计算机太晚了，科班出身的已经学四年了，你能一下赶上吗？虽然我觉得他说的有道理，但是我总觉得只要努力还是有可能赶上的。不管怎么样，我都要努力尝试一下，不能放弃。

于是从书店买来谭浩强那本《C语言程序设计》，就这样开始自学编程。把书上的程序例子在电脑上都跑一遍，C语言掌握得差不多了，然后报名参加了计算机三级考试。

四处应聘，转行成功

1999年7月，我觉得自学编程差不多了，开始应聘一些程序员岗位。这时候我最想去深圳，因为深圳比珠海更有发展。所以开始投的主要是深圳的公司，但由于我没有计算机专业文凭，也没有程序员工作经历，投的简历要么没有回音，要么笔试没有通过。

深圳去不了，开始考虑深圳附近的东莞。一个周末，我去应聘东莞一家电子厂，现场笔试C语言程序设计和汇编语言程序设计。我的C语言程序设计顺利通过，但是汇编语言没有学过，自然没有通过。

因为我很想去这个公司，而且我发现这个公司几乎每个周末都招聘，于是回到珠海买了本单片机原理的书，很快学会了C51汇编语言编程。1999年10月，再一次前来应聘。这次换了一个考官，由于之前有准备，汇编语言考试也顺利过关。但考官一直在犹豫是否录用，因为我没有计算机专业文凭，也没有程序员工作经历，仅仅这个笔试过了能说明什么呢？这个时候我对他说，我的数学基础好，有做程序员的潜力。考官还在犹豫，我又对他说，如果给我工作机会，两三个月内工作不满意，可以辞退我，不用支付任何工资。或许是程序员的需求太旺盛，或许是自己的诚意打动了他，我被录用了。

我终于转行写程序了，而且这个过程并不容易。但是我在后来的工作中发现，有很多像我这样从非计算机专业转行做软件开发的，他们的转变似乎没有我这样麻烦。因为他们很多都是由同学朋友介绍，或者在别人的帮助下自学的，这比我靠自己一个人自学更为有效。总之，如果有一个人，或者有一个环境帮你学习，比你自己一个人学习，效果肯定更好。

从珠海到东莞，从化验员到程序员，对我来说，是一个非常重要的转变。因为我擅长数学，写程序比做化验更能发挥自己的长处。另外，从行业发展空间的角度来考虑，已经发展成熟的化学行业的发展空间也无法和方兴未艾的IT行业的发展空间相比较。

琢磨考研：我需要新的成长空间

虽然这个公司录用了我，我已经转行做程序员，但是我一直很担心被炒鱿鱼，因为除了自学编程，我没有实际工作经验，也就一直不敢怠慢。住在工厂里，晚上或周末有空就主动去加班学习，看别人做的东西，自己也主动做一些。这其中除了担心被炒以外，我还想提高自己的水平，然后跳槽去了深圳。

两次跳槽

在东莞工作了半年之后，也就是2000年5月，我跳槽到了深圳蛇口，在一家日资电子公司做软件开发。我在珠海的同事一个月工资才一两千，我说这里一个月有五千，他们都不相信。

2000年11月，我又跳槽到深圳另一家公司，一直待到2006年。刚来这个公司的时候，我的发展还比较顺利，半年后就当上项目小组长，也是同一批进公司的人当中第一个当项目组长的。但是在接下来的几年，我在公司再也没有提升，因为这几年公司几乎没有发展，员工自然也没有多少发展空间。可见，选择一个有成长空间的公司，多么重要！

我在这个公司待了很久的一个重要原因，就是工作还算轻松，不用加班，



这样的环境适合考研。按照自己最初规划，本科毕业先工作，看看什么工作适合我。现在我已经发现软件开发最适合我，而且已经积累了一点工作经验，显然下一步的打算就是考研。更何况我自己不是计算机专业出身，知识不够系统；本科文凭在贬值，读研一年比一年热。虽然我在公司里比较受重视，工资也比我很多同学高，但我时刻提醒自己：要有忧患意识，要考虑长远一点。

考研•读研

2002年，我第一次报名考研，报考北京航空航天大学计算机专业。一边上班一边复习，现在想起来，虽然白天上班累一点，只要充分运用晚上的时间，是完全可以考上的。问题是自己没有好好努力。在接下来的几年里，我每年都会报考研究生，但是每年都没有认真复习，结果可想而知。

一转眼就到了2005年。由于公司发展不好，我还是没有什么发展空间，我想换一个工作环境。这时想到了北京，因为北京有很多大学和研究所，学习机会多，我还想一边工作一边学点什么。考研像是一块石头，一直在心里没有完全放下。

这一年我已经28岁，很多人都觉得考研太晚，不值得。但是我仔细考虑，觉得还是要考研：（1）考研一直是我的一个梦想，是梦想就要努力追求；（2）读研属于一种教育投资，教育投资回

报周期很长，但其回报往往在更长的时间里都会有体现。

这时我刚知道北大有了软件学院，主要面向工程和实际应用的，比较适合已经工作很多年的人。但我报名的时候还是选了北大计算机系，因为计算机系更难考，我想看看自己的水平，另一方面，如果没有考上计算机，还有调剂到软件学院的机会。又一次考研复习开始了，这次复习比往年认真很多。还是白天上班，晚上复习。

回深圳考完研后我就辞职了。不管考研成绩如何，我都决心来北京了。成绩出来，我考了367分，复试分数线366分。本以为自己的工作经验可以在复试中发挥作用，但是复试根本没有问这些，加上自己的一些失误，第一志愿复试没有过。考虑到自己工作很多年了，去差的方向或者去深圳研究生院学习三年似乎不合算，就调剂到软件学院了。

这次考研，虽然也不是非常完美，但是基本达到预期目标。那时候也想，要不要再考一年，但思考过后，还是打算去读研了。因为不能花费过多的时间代价，去寻找最完美的结局。而是要在尽可能快的时间里，做出比较完美的选择。

2006年9月，也就是本科毕业八年之后，我辞了以前工作，又重新回到校园。这时候29岁，年龄大一点，有时候真的有压力，甚至还有一些顾虑。所以和这些小好几岁的弟弟妹妹们相比较，我更加珍惜学习机会。

和读本科时一样，刚刚入学，我就在仔细考虑，我在读研这个阶段要怎么度过，我要选择一个什么样的发展方向。在选择自己发展方向的时候，我综合考虑了过去、现在、将来三个因素：（1）过去几年的工作经历是一种财富，要尽量结合运用，不要放弃；（2）现在学校在这个方向有好的学习环境，这样能最快地提高自己；（3）以后这个方向要有发展空间。

后来我选择了在嵌入式设备上做多媒体算法研究这个方向。因为，过去几年的工作，做过一些音频多媒体算法优化工作，并且深深感受到这是很多企业的关键技术之一；而且学校在这个领域有很好的条件，有相关的课程和实验室。

选择好适合自己的发展方向之后，我就一直围绕这个方向选课、做项目、实习。研究生第二年，我们开始实习，根据自己的发展方向，我投了一些招聘多媒体算法研究的公司，后来到了汤姆逊北京研发中心，主要从事JPEG2000的算法研究。其实这个时候，我还在北大另外一个实验室做DSP上的无线视频监控项目。一个时间同时做两件事情，听起来很好，其实不然。因为一个人的精力有限，同时做两件事情，往往不如集中精力做好一件事情。

我一直很庆幸自己主动选择了改行做软件开发。做自己喜欢的、擅长的，工作就成了享受，创造奇迹的机会会大得多。

再度求职：找到适合自己的岗位

一转眼到了研三，要找工作了。仔细看了网上很多关于找工作的帖子，我开始思考以下问题：（1）我要做什么性质的工作？技术，管理，还是销售？（2）我要选择什么样的公司？

经过以前一位同事的推荐，我选定了一家国内知名的芯片设计公司，工作是做芯片上多媒体算法优化，公司和岗位都很适合我，待遇也很不错，还能解决北京户口。从2008年10月开始实习，到现在快一年了，马上就要负责一个小项目了，这对我今后转型走“技术+管理”的路应该是一个很好的机会。

在实习和工作的过程中，我自己慢慢积累了一些工作方法。

1. 做好计划与总结。说远点，以后10年、20年要做什么；说近点，这个月、这个星期、今天要做什么，都要有明确的计划，并且写在自己的Google记事本上。然后定期总结回顾，看这段时间的计划执行效果如何，以后有什么需要注意或者调整的。

2. 开拓自己的视野。充分运用丰富的网络资源，订阅《南方周末》、《华尔街日报》、Google黑板报、CSDN等专业或非专业的资料。要做笔记，以便于以后查阅。

3. 选择的权衡。工作过程中会面临很多选择，要尽量选择对公司最有价值的工作，尽量选择最能发挥自己特长的工作，经常想想公司或者部门现在最缺少什么，如果我能填补这个空缺，对公司和个人的发展都有好处。做工程很多时候不能过分追求完美，而是在尽可能短的时间里做出满足要求的产品。

4. 正确的态度。积极主动地工作、学习、沟通、交流。

结语：不断总结，永远沟通

客观地说，这几年里，我一直没有闲着，转行、考研、工作。虽然没有什么大的成绩，但是一直在努力。在竞争日益激烈的今天，仅仅靠努力还不够，还要注意方法，所以要不断总结经验，吸取教训。

- 多向别人学习，多向前人借鉴。一个人考虑问题总有局限，多学习别人的优点，多借鉴前人的总结，可以少走很多弯路。最重要的是多和别人交流，书本和网络，都不能取代人和人之间的面对面交流。
- 认准了一件事情，要全力投入，甚至还要勇于放弃一些眼前利益。
- 凡事有计划，认真执行计划，勤记录，常总结。要把自己想学的东西、想做的事情制订出一个计划，还要把自己工作、学习、生活的感受记录下来，要经常总结，不能虚度光阴。
- 除了好好工作学习，还要学会生活。开心的生活、健康的身体、幸福的家庭都是工作学习的有力保障。

作者简介



黄小明，1977年生于安徽安庆。现就职于北京中星微电子有限公司。

责任编辑：李雨来（liyl@csdn.net）

代码猴子与童子军军规

■ 文 / 韩磊

2007 年 3 月， 我 在 SD West 2007技术大会上聆听了Robert C. Martin（Uncle Bob） 题 为 Craftsmanship and the Problem of Productivity: Secrets for Going Fast without Making a Mess的主题演讲。休闲打扮的Uncle Bob，以一曲引起全场共鸣的“Code Monkey”（代码猴子）开场，介绍了测试驱动开发的种种好处。回来一查，所谓“代码猴子”，是编程者自嘲，也是讽刺低水平开发者的一种说法。

是的，我们就是一群代码猴子，上蹿下跳，自以为领略了编程的真谛。可惜，当我们抓着几个酸桃子，得意洋洋坐到树枝上，却对自己造成的胡乱熟视无睹。那堆“可以运行”的乱麻程序，就在我们的眼皮底下慢慢腐坏。

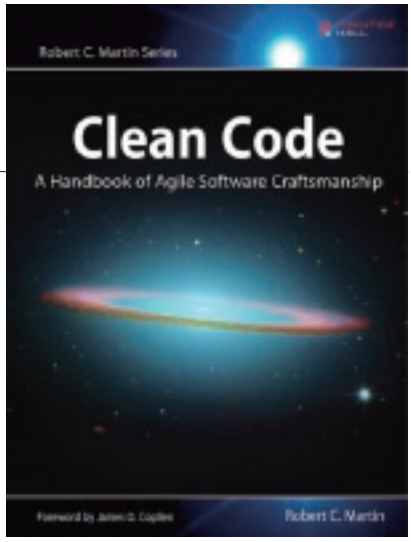
从听到那场以TDD为主题的演讲之后，我就一直关注Uncle Bob，还有他在TDD和整洁代码方面的言论。去年，人民邮电出版社计算机分社拿一本书给我看，封面上赫然写着Robert C. Martin的大名。这本书名为Clean Code（《代码整洁之道》），乃是Object Mentor（Uncle Bob开办的技术咨询和培训公司）一干大牛在编程方面的经验累积。按Uncle Bob的话来说，就是“Object Mentor整洁代码派”的说明。

接触开发技术十多年以来，尤其是从事IT技术媒体工作六年以来，我见过许多对于代码整洁性缺乏足够重视的开发者。不算过分地说，这是职业素养与基本功的双重缺陷。我翻译《C#编程风格》（已由人民邮电出版社出版）和《代码整洁之道》，实在也是希望在这方面看到开发者重视度和实际应用的提升。整洁的代码未必一定高效、强固，但高效、强固、可扩展和可维护的代码大抵是整洁的。有一些开发人员认为，维持代码整洁意义不大，因为需求变化是如此之快，当前项目很快就会被抛弃。其实，正是由于需求变化，整洁代码才更加显得弥足珍贵。且不论后续扩展，使用敏捷开发手段、不断跟随需求变化迭代重构，当代码变得越来越整洁，应用也就趋向于完成需求目标。反之，项目也会随代码的混乱而滞后甚至崩塌。

正如James O. Coplien在序中所言，宏大建筑中最细小的部分，比如关不紧的门、有点儿没铺平的地板，甚至是凌乱的桌面，都会将整个大局的魅力毁灭殆尽。这就是整洁代码之所系。Coplien列举了许多谚语，证明整洁代码的价值。整洁代码的重要性毋庸置疑，问题是如何写出真正整洁的代码。

整洁代码的实施手段不外乎测试先行与迭代重构。在软件变得越来越复杂的时候，没有人可以从一开始就写出且一直保持写出无须更改的绝世好代码。软件从建筑学获益良多（架构、工程……），但借喻总归是借喻，何况还有来自制造业的或其他行业的借喻（精益、全员生产维护……）。所有的借喻加在一起，都无法覆盖或取代软件生产（又是借喻！）本身的本质特征与实施手段。软件足资敬畏，Uncle Bob及其同伴是在以一种谦恭而求实的姿态，探寻软件开发的特征与实施手段。相对于争论工程与工艺孰为正宗，这样的做法无疑更为务实，也更具指导意义。

本书既是整洁代码的定义，亦是如何写出整洁代码的指南。Uncle Bob认为，“写整洁代码，需要遵循大量的



小技巧，贯彻刻苦习得的‘整洁感’。这种‘代码感’就是关键所在。……它不仅让我们看到代码的优劣，还予我们以借戒规之力化劣为优的攻略。”作者们阐述了在命名、函数、注释、代码格式、对象和数据结构、错误处理、边界问题、单元测试、类、系统、并发编程等方面如何做到整洁的经验与最佳实践。长期遵照这些经验与实践规则编写代码，所谓“代码感”也就自然而然滋生出来。

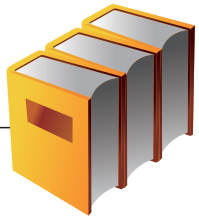
书中更有价值的部分是Uncle Bob本人对三个Java项目的剖析与改进过程的实操记录。通过这多达三章的重构记录，Uncle Bob充分地证明了童子军军规在编程领域同样适用：离开时要比发现时更整洁。

在《代码整洁之道》结束语中，Uncle Bob提到别人给他的一条腕带，上面的字样是Test Obsessed（沉迷测试）。Uncle Bob“发现自己无法取下腕带。不仅是因为腕带很紧，而且那也是条精神上的紧箍咒。……它一直提醒我，我做了写出整洁代码的承诺。”有了这条腕带，代码猴子成了模范童子军。我想，每位开发者都需要这样一条腕带吧？

（《代码整洁之道》一书即将由人民邮电出版社推出。）

■ 责任编辑：李雨来（liyl@csdn.net）

新书上架



Google Apps应用完全指南



原书名： Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop
作者： [美] Scott Granneman
译者： 李强 孙向阳
出版社： 机械工业出版社

Google Apps提供了很多业务和有效软件，其中部分是免费的，部分可用极低的价格得到。开始学习Google Apps时往往感觉很容易，但要想充分地利用好它则需要专业的指导。

书中介绍了如何应用Google Apps这套“云计算”的应用方案，来完成企业、学校或团体的通信和协作。全书共分7个部分21章，分别介绍了Google Apps的注册；E-mail、Contact和Calendars到Google Apps的迁移；Gmail、Google Calendars、Google Docs和Google Site的应用；其他服务和注意事项。作者根据帮助WebSanity公司使用Google Apps的丰富经验，介绍了许多在其他地方找不到的技巧和窍门。

细说PHP

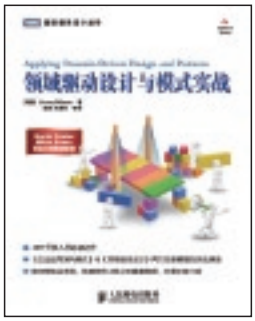


作者： 高洛峰
出版社： 电子工业出版社

该书是易第优教育为推广普及LAMP技术而推出的系列丛书之一，以实用为目标，系统而图文并茂地介绍了PHP的相关技术及其在实际Web开发中的应用。

全书共17章，每一章都是PHP独立知识点的总结。内容涵盖了PHP的运行环境搭建、Web服务器Apache的配置与应用、动态网站开发的前台技术、PHP编程语言的语法、PHP的常用功能模块和实用技巧、MySQL数据库的设计与应用、PHP 5面向对象的程序设计思想、Web开发的设计模式，以及包含DIV+CSS、mysqli扩展模块、数据库抽象层PDO、Smarty模板技术等目前PHP开发中最主流的技术。最后以一个比较完整的、采用面向对象思想，以及通过MVC模式设计，并结合Smarty模板的CMS系统为案例。

领域驱动设计与模式实战



作者： [瑞典] Jimmy Nilsson
译者： 赵俐 马燕新 等
出版社： 人民邮电出版社

大多数开发人员对领域驱动设计、模型驱动开发、模式等术语并不陌生，可要在实际开发工作中熟练运用却并不容易，因为这些概念本身非常抽象，不易理解且难以运用。但实际上，如果能够很好地运用模式、领域驱动设计和测试驱动开发，则可以赋予架构师和开发人员前所未有的能力，使他们能够创建功能强大、健壮且可维护的系统。

作者Jimmy Nilsson是一位有着超过20年从业经验的资深软件架构师。他运用多年的丰富经验，将抽象的概念运用到实际示例中，真正在抽象理论与实践运用之间架起一座桥梁。同时，他将Martin Fowler《企业应用架构模式》和Eric Evans《领域驱动设计》两部经典名著中的思想精髓以及重构、测试驱动开发等技术融会贯通，通过大量C#实例加以阐释，跨越了领域模型、数据库与UI层之间的障碍，真实展示了创建高质量的企业级应用架构的全部过程。

程序员羊皮卷

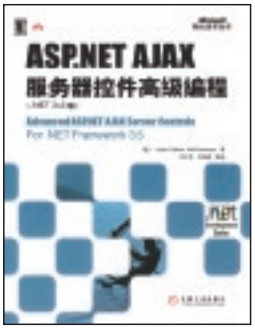


作者： 张大志（LEO）
出版社： 电子工业出版社

年轻人走入社会的时候都会发现一件事——世界掌握在年纪比自己大的人手中，这些人制订并实施着规则。因此，很多人越来越循规蹈矩，但却心里愤懑：没有自己的思想，没有快乐的工作，甚至失去了自己的灵魂，怎么办？

仔细考察程序员职业生涯的各个阶段，大致都会遭遇：求职不顺利、工作中困难当道、无法解决职场冲突与矛盾、不知道自己的职业目标是什么、无法有效地调整行动向目标前进，等等问题。本书作者以自己12年职场阅历的所见所闻，对以上问题提供了思路与方案。书中内容涵盖了如何读大学、怎样求职、怎样应对职场政治，以及技术人员创业等各个方面，以程序员的成长为主线，讲述各阶段所遇到的困难、问题以及相应的解决方法。书后还讲述了程序员职业规划九大工具箱，以具体的操作方法来帮助人们看清前行之路。

ASP.NET AJAX服务器控件高级编程



原书名：Advanced ASP.NET AJAX Server Controls:For .NET Framework 3.5
作者：[美] Adam Calderon；Joel Rumerman
译者：罗江华 罗恒峰
出版社：机械工业出版社

这是第一本针对有经验的ASP.NET开发人员所编写的、用于实现自定义ASP.NET AJAX服务器控件的书籍。不同于其他ASP.NET AJAX图书，此书专注于服务器控件的开发以及在ASP.NET 3.5 AJAX中所展现出来的效率提升，并且探讨了如何在最新的Visual Studio 2008中加快开发AJAX的特性。

书中重点讲述了如何使用ASP.NET AJAX来创建具有AJAX功能的服务器端控件，通篇都结合ASP.NET AJAX的三个方面：Microsoft AJAX库、ASP.NET 2.0 AJAX 扩展以及ASP.NET AJAX 控件工具箱来展开讨论。全书共分四个主要部分：第一部分聚焦于Micosoft AJAX库和JavaScript语言。第二部分主要讲述创建分布式的启用AJAX的控件。第三部分介绍通信，阐述了客户端控件与服务器端交流时的不同之处。第四部分则重点介绍了ASP.NET AJAX控件工具箱的相关知识。

该书作者之一Adam Calderon是InterKnowlogy公司应用程序开发实践团队负责人，他从Alpha版本开始对ASP.NET AJAX进行研究，并且将这项技术率先应用到Atlas访问程序中。另一作者Joel Rumerman是高级.NET开发人员，他从ASP.NET AJAX代码代号为Atlas发布以来一直在进行专门研究。两位作者致力于深入ASP.NET AJAX框架的内部工作和扩展性。

项目管理修炼之道



作者：[美] Johanna Rothman
译者：郑柯
出版社：人民邮电出版社

在步伐越来越快的今天，如何降低风险，做好管理，保证项目成功，是一个令许许多多人寝食难安的话题。这本荣获软件业奥斯卡——Jolt大奖的优秀图书，直面难题，提出了自己的思路和解决方案。

本书作者Johanna Rothman是世界知名的管理顾问，擅长高科技产品开发管理。她多年来帮助许多高科技公司成功地解决了各种有关产品开发管理的棘手问题，本书正是她宝贵实战经验的提炼。书中从应对实际风险的角度出发，讲述了从项目启动、项目规划到项目结束的整个管理流程；展示了作者的完整思考过程，从评估项目背景、选择生命周期，直到为项目建立清晰的条件；同时穿插了丰富的提示和真实案例，介绍了可能遇到的常见问题。这些真知灼见不仅适用于软件项目管理，同样适用于其他产品的开发项目。

艾丝特•德尔碧联合有限公司总裁评价说："每个项目都是独一无二的，因此所有的项目经理都得多掌握几种管理项目的方式。Johanna给我们展示了她的思考过程，从项目背景的评估，到生命周期的选择，直到为项目建立清晰的条件。她的建议能够帮你作出明智的决策，使项目走向成功。"

这本可使项目经理即刻上手的实战指南，适合于任何类型的项目经理：无论你是使用瀑布式、迭代式、还是敏捷式生命周期模型，都可反复研读，从中得到有益的提示和帮助。

全球排行榜

	Amazon
1	Crush It!: Why NOW Is the Time to Cash In on Your Passion by Gary
2	How to Survive the End of the World as We Know It: Tactics, Techniques, and Technologies for Uncertain Times
3	Coders at Work
4	Mac OS X Snow Leopard: The Missing Manual
5	PMP Exam Prep, Sixth Edition: Rita's Course in a Book for Passing the PMP Exam
6	Halo Encyclopedia
7	Windows 7 Secrets
8	Inbound Marketing: Get Found Using Google, Social Media, and Blogs
9	Windows® 7 Inside Out
10	Beginning iPhone 3 Development: Exploring the iPhone SDK

	天龙书局（台湾）
1	Silverlight 3.0 全面精通手冊——使用 VC#
2	Google Android SDK 开发范例大全
3	大话设计模式
4	走出软体工场
5	Facebook 与 funP 应用程式设计
6	jQuery 开发实战（Learning jQuery 1.3）
7	Google！ Android 手机应用程式设计入门，2/e
8	C++ Primer，4/e 中文版
9	Linux Shell 程式设计实务
10	Windows Device Driver Programming 驱动程式设计

	第二书店
1	构建高性能 Web 站点
2	学习 OpenCV（中文版）
3	算法导论（原书第 2 版）
4	编程之美——微软技术面试心得
5	程序员修炼之道——从小工到专家
6	正在爆发的互联网革命
7	细说 PHP
8	我是一只 IT 小小鸟
9	Head First 设计模式（中文版）
10	黑客攻防技术宝典：Web 实战篇



Sundries CONTRAST EHP-OH100

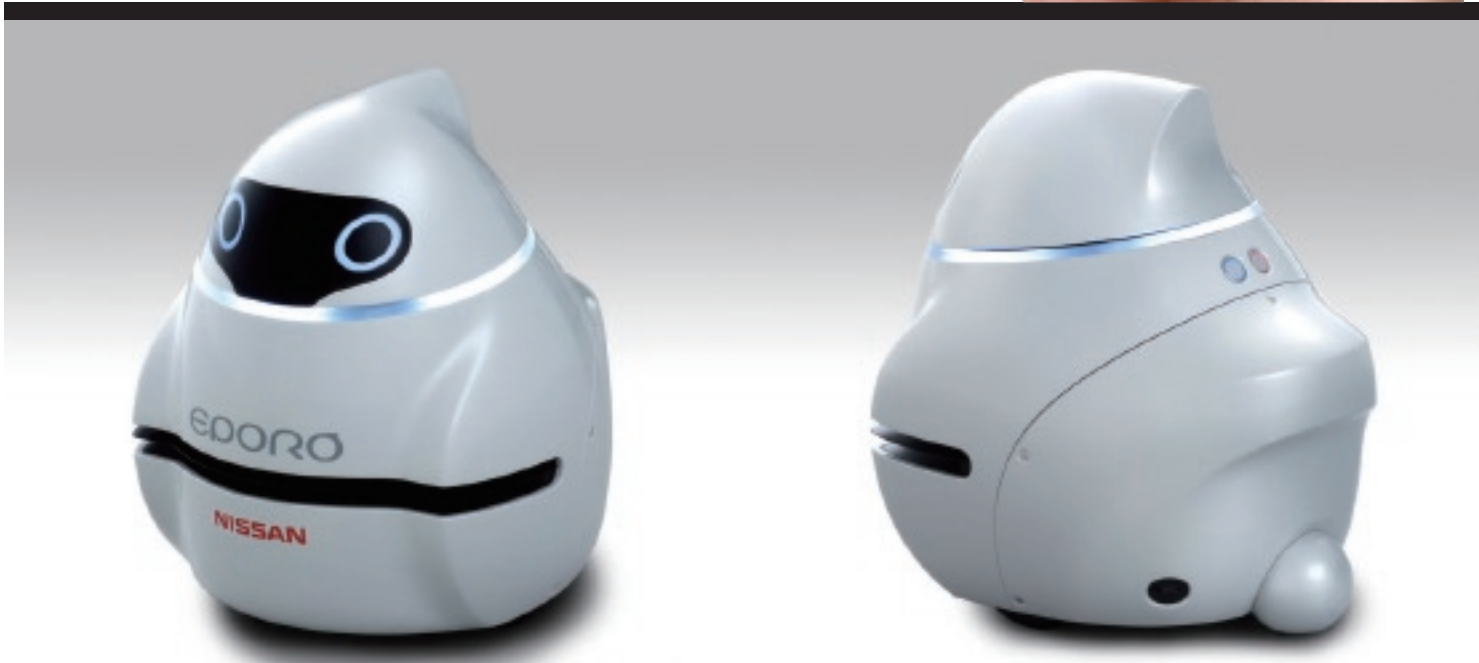
入耳式耳机对听力影响不小，而头戴式耳机携带起来却不甚方便。ELECOM推出的这款折叠头戴式耳机在不需要佩戴的时候，只要简单几下即可将耳机折叠起来，放在包里也不会占用太多空间。

GEEK 产品

Wikireader

曾经有人测算过，如果将Wikipedia的内容印刷出来其厚度可能要有半人高。随着Amazon Kindle这类电子阅读器的产品的出现，如今也有公司打起了Wikipedia的主意，要将其搬到移动终端上来。如果有机会的话，你是否会想配备一个呢？





Nissan Eporo

Nissan的工程师们在东京举行的Ceatec展会上展示了其最新的鱼形小车机器人Eporo。像鱼群一样，Eporo机器人也可以向同伴分享它自己的位置和其它信息，避开障碍。Eporo安装了LRF雷射测距系统，可以扫瞄到前方288度范围内的障碍以避让并通知同伴。未来这项技术有可能用在城市拥挤的交通中，避免交通意外的发生。



Air Multiplier

如果说左边这个东西是电扇，你可能不信。最近Dyson发表了一个新的技术，名为“Air Multiplier”，最大的特色就是号称“无风扇”的设计，其运作原理是靠底座吸入空气，并通过环形的增压器，从只有1.3mm的孔隙“挤”出空气，现在Air Multiplier这个技术被Dyson拿来做电风扇。

iida Polari

iida Polaris是一款类似于小机器人设计的概念手机装置，新颖小巧，超级可爱。这款手机除了具备基本的通话功能外，最主要的优势就是为用户提供行程安排，GPS导航，RSS订阅等服务功能。同时还具备主动学习的功能，可以通过观察学习主人的生活习惯以及了解主人的兴趣爱好，同时通过电脑分析数据，设置自身的最佳状态以适应使用者的需求。





MindFLEX

在CES 2009大会上美国Mattel公司推出了一款非常特别的游戏MindFLEX。你需要戴上配套的耳机，然后集中注意力，耳机会接收你的脑电波，感应器会将信号反应到下面的气流控制器上，让那个紫色的小球飘浮在空中。游戏的目标是用意念控制小球的高低飞过各种小障碍。

新产品&新工具

AminNes

AminNes是一款基于Flash使用ActionScript开发的Nes模拟器，目前已经实现绝大部分硬件模拟，至少有500个以上的ROM可以正常运行。目前该模拟器已经由作者开源。AminNes在Flash性能方面融入了作者多年的编程经验，进行反复优化及改进，整体运行速度流畅。到后期对文件、程序结构上也进行了多次调整及组织，达到简易明了的可阅程度（含有中英双文注解）。对于ActionScript和Flash优化以及相关内容感兴趣的用户来说，是份很好的参考素材。



Adobe Flash发布平台

对很多公司来说，把软件发布到软件商店里是唯一的发行渠道。Adobe近日宣布的Flash发布平台服务就是为了解决这个问题而推出的。该服务提供免费和付费两种方式在社交网络上发布应用，同时也提供基于社会网络和传统方式结合的数据跟踪和分析功能。

Microsoft Security Essentials 1.0.1611.0 简体中文正式版

微软之前推出的免费安全防护软件Microsoft Security Essentials近日发布了正式版，该产品标志着微软正式进入杀毒领域，同时也替下了之前不冷不热的Windows Live One Care。 Microsoft Security Essentials 可直接从微软网站下载，安装简便，没有复杂的注册过程和个人信息填写。

用友“T系列”智慧型中小型企业管理软件

10月16日，用友软件在北京面向中小企业用户正式对外发布新一代“T系列”智慧型中小型企业管理软件。这是中国首套完全基于SOA架构的中小企业管理软件。本次发布的用友新一代“T系列”智慧型小型企业管理软件，包含了用友T1、T3、T6三个系列产品。分别满足个人或小企业某个特定应用需求的单项级应用、企业中某个部门或某几个部门管理需要的部门级应用、企业整体运营管理需要的企业级应用。

Paint.NET v3.5 Beta 2



如果你觉得Windows自带的画笔功能太差，又不想使用重量级的Photoshop，那么可以试试Paint.NET。Paint.NET是一个图像和照片处理软件，它由华盛顿州立大学的学生开发和维护并由微软公司提供项目指导。早期定位于Windows画笔的免费替代软件，现在逐渐发展为一个功能强大且易用的图像和照片处理软件。支持图层、无限制的历史记录、特效和许多实用工具，并且是开放源代码的。此次的版本修正了一些Bug，并提升了多图层图片切换时的性能。

JIRA 4

JIRA是一个获得过Jolt大奖的集项目计划、任务分配、需求管理、错误跟踪于一体的商业软件。融合了项目管理、任务管理和缺陷管理，许多著名的开源项目都采用了JIRA。新版本提供了OpenSocial组件，改进了Dashboard，JIRA查询语言（JQL）支持强大的查询和过滤功能，此外还包括超过950个其他特性和改进。



Oracle Database 11g R2

老牌数据库厂商Oracle在2009年9月底发布了Oracle 11gR2。Oracle 11gR2引入了很多新特性：Oracle RAC提供了即插即用型网格和新的服务器池功能，让组织可以减少服务器成本；单节点Oracle RAC，这个新的解决方案可以让用户对网格中非关键数据库环境进行合并。Oracle 11g R2高级压缩和分区支持，通过提供高达2~4倍的压缩比降低存储成本。Oracle 11g R2自动存储管理（ASM），防止服务器和存储设备在等待故障时空闲。Oracle 11g R2与Oracle数据库机结合可提供高达10倍的查询性能。

Google App Engine Java SDK 1.2.6

Google App Engine 本月发布了用于Java的SDK的新版本，增加了接收邮件的处理接口，新的AllocateIDs方法、用于任务查询API的DatastoreQueues，修改了默认的队列URL以适应Python，新的Appcfg.sh action(cron_info和update_queues)，除此之外，服务器管理界面也进行了修改，默认启用ssl-enabled选项。

Windows Phone

10月15日，微软公司携手90余家移动通信领域的合作伙伴宣布将推出首批全新的Windows Phone智能手机。Windows Phone手机将采用最新的Windows Mobile 6.5操作系统，提供更加完善和易用的用户界面和更为丰富的浏览体验。此外，特别为中国用户本土化定制的My Phone服务也同时推出，让中国用户可以轻松而安全地将联系人、日程、短信以及照片、视频等重要信息与网络和PC实现同步。

Zend Studio 7.0.1

Zend Studio for Eclipse目前是PHP官方的开发工具，同时也是很多PHP开发者的首选工具，它基于Eclipse 的最新版本构建(Galileo)。在新版中增加了对PHP 5.3的支持；增强了Zend Framework的集成；增强了代码编辑功能，支持即时代码重构；更好地集成了Zend Server用于快速错误分析和调试；同时改进了整体性能。

ASP.NET MVC 2 Preview 2

新版本的ASP.NET MVC在ASP.NET 3.5运行时上提供了一个新的模型视图控制器（MVC）框架。这意味着开发者可以通过MVC方式开发各种Web应用。新的ASP.NET MVC框架选项非常干净，而且容易使用。它将允许你在你的应用中很轻易地保持关注分离，而且有助于进行测试和TDD。Preview 2在之前版本的基础上增加了客户端验证；精简了针对Area支持，以及抽象的数据标记（Data Annotations）验证和元数据提供者等功能。ASP.NET MVC 2 Preview 2可与MVC 1共存，但在安装MVC 2 Preview 2之前必须先卸载Preivew 1版本。



张晓东，2002年毕业于北京航空航天大学，有7年Java企业级应用开发经验，目前是奥博杰天软件公司TEC-Evangelist的一员。

软件的性格

人有很多种，有的细腻，有的粗犷，有的宽容，有的苛刻，有人自信，有人友善，有人专注，有人博学，有人充满创意，有人富于远见，有人学院气息十足，有人则是满脑子利益与金钱。就是这样一些林林总总，有血有肉的IT精英们，聚集在风格各异的软件公司之中，以智慧为原料，以创造为动力，制造着世上精彩纷呈的软件产品。

■ 文 / 张晓东

众所周知，软件制造过程95%以上是脑力劳动，发生、发展、成熟都在设计人员和工程师的脑海之中。自然而然，这些“软”产品就先天地被打上了创造者的烙印。而在当今这个软件业高度产业化的时代，软件的特质已经从个人烙印逐渐转变为团队的烙印，甚至公司的烙印。本文便尝试以当前软件业的佼佼者——微软，谷歌，Adobe，苹果和IBM为蓝本，探究一下这些公司及其旗下产品的个性特点。

微软

业内最精明的企业，凭着在个人计算机软件市场上一家独大的局面，成就了大名鼎鼎的前全球首富比尔·盖茨，21世纪最后那个十年这位盖茨先生一度是全世界高科技行业新贵的成功典范。微软奇迹般地把旗下一系列颇有些华而不实的产品做得声名远播，有声有色。Windows，MS Office，SQL Server，Visual Studio，Hotmail以及Windows Live套件可谓世人皆知，而且大部分价值不菲。

Windows：这个占据全球个人电脑市场90%以上份额，以简单和界面友好著称的操作系统家族，每一次推陈出新都会引来用户的无数埋怨。如今，面对着几天来那么一次甚至一天要来几次的带着触目惊心的警示信息的补丁，笔者从恼火到愤怒终至无奈。微软聚集了数目庞大的软件精英，完成的核心产品为何如此的差强人意？！如果有人说Windows在市场上的成功自有它的道理，不妨看看盖茨先生1998年在华盛顿大学道出的个中真谛“我希望他们盗版的时候选择我们的软件，等他们上瘾之后我们会在下一个十年里找办法对付他们”。

Office很出色，Word、Excel、Powerpoint、Outlook集成得非常完美，不过如果我们掂量一下它的价格以及打包销售给消费者的诸多可有可无的功能，窃以为，如果不是仰仗着“放纵盗版外加捆绑销售”的市场策略，Office的销售在中国不会这么火。

Visual Studio，价格昂贵的著名Windows开发套件，笔者满怀期待的试过其精简版Visual Studio C# Express 2008。难以置信的是，在21世纪的软件开发行业，这个几百兆的庞然大物居然完全不支持日常工作中最常用的“Open Type/Class”，“Organize Import/using”，这对于被Eclipse等高效开发工具宠坏了的程序员来说，真是太蹩脚了。更不可思议的是，对于当代IDE必备的重构功能，VS C# Express也只支持两种——“重命名”和“提取方法”。总而言之，这是笔者用过的效率最低的源码开发工具之一。

作为一个使用了微软产品逾10年的老用户，面对着这一系列华而不实的产品，笔者真心希望微软公司在产品的成熟度和实用性方面有所改观。同为企业的灵魂，产品和市场，如果能够更加均衡的发展，应该可以走的更远。

谷歌

这个世界青睐不断创新的年轻人，所以这个时代成就了谷歌。凭着出类拔萃的搜索引擎，谷歌走向了世界。它被很多网友亲切地誉为“神的主页”。

搜索引擎并未让这位年轻人止步，短短的几年时间，谷歌推出了一系列充满创意的产品。Gmail的垃圾邮件过滤能力让人称道，贴心的邮件会话视图更让人爱不释手。谷歌地图街景模式生动细致，拖拽式的线路调整让

人耳目一新，集成网络摄像头的想法也天马行空般的新颖。可定制的门户iGoogle是 Web2.0的宠儿，页面外观、布局、内容，通通用户说了算。谷歌Picasa图片自动组织功能方便轻巧，编辑功能中的“手气不错”把腐朽化为神奇就在鼠标那轻轻的一点，网络相册离线管理更是让图片的分享轻松简单。谷歌 Chrome浏览器，地址栏可以一栏多用，为网站建立快捷方式，崩溃控制外加隐身模式。谷歌地球，可以带您免费遨游太空，足不出户地感受如何在遥远的城市穿街走巷，翱翔天际而俯瞰人间万象。

Gmail的会话视图，iGoogle的门户，Picasa的照片管理，Chrome的轻松写意，用过之后常常感慨，软件难道本来不该如此么，为什么其他软件没做到呢？

Gmail的会话视图，iGoogle的门户，Picasa的照片管理，Chrome的轻松写意，用过之后常常感慨，软件难道本来不该如此么，为什么其他软件没做到呢？

Adobe

Adobe是一个软件业的多面手，涉足的领域，无论动画特效、文档制作、图片处理还是Web 2.0时代的AIR开发平台，技术始终如一的卓越，产品无一例外的精致。

作为网络可打印文档的事实上的标准，Adobe创造的 PDF格式，凭着优异的算法，以小到惊人的尺寸，为文字、表格和图片的阅读和打印提供着超清晰的体验。相比于微软对于 MS Office文件格式的三缄其口，秘而不宣，Adobe将 PDF格式公诸于众并且允许开发者免费使用的创举可以说是尽显技术领先者的大家风范。

作为交互式矢量图和 Web动画的标准，Flash巧妙的将矢量图形运用于移动图形的表现，既极大地压缩了动画特效所需要的文件尺寸，又高度简化了动画的制作过程，在全世界催生出一代“闪客”，短短几个月里就将带宽还很有限的早期互联网从单纯的信息载体和电子公告牌转变为生动无比的娱乐新媒体。而后随着 Flash MX的推出而发展而来的 FLV（Flash Video）流媒体格式更是凭着 CPU占有率低、视频质量良好以及只有普通视频文件1/3体积的诸多优点在网络上盛行并博得新浪播客、优酷、土豆、YouTube等主流视频网站的集体眷顾。

秉承着“只有想不到，没有做不到”的精神，Adobe不屑于让人眼花缭乱的市场宣传，不盲目发展五花八门的产品线，以但凡涉猎必出精品的气魄，赢得了同业者的赞叹。

苹果

业内独一无二的浪漫主义艺术家，把功能或简单或复杂的IT产品作为艺术品设计并且作为时尚消费品销售的天才。

在这个高度信息化的时代，从享受音乐、游戏娱乐、沟通联系到商务办公，消费者对于IT产品的强劲需求是毋庸置疑的，但是这些本质上构筑在二进制算法之上的IT产品却曾经因为过分专注于强劲的功能和对外观的忽视而将大众消费者拒之门外。苹果以一双慧眼捕捉到了这个机遇并用一双巧手改变了这一切——凭着精美、易用和创新的设计理念，苹果推出了一系列精巧而浪漫的产品并成功地将它们销售到世界的各个角落。

苹果公司 COO Tim Cook在与金融分析师的一次电话会谈中透露，自诞生之日起截至 2009年 4月，仅 iPod Touch和iPhone就已售出逾3700万台，这让人瞠目艳羡的数字背后，是苹果将艺术和浪漫气息引入 IT业后应得的尊重和回报。

IBM

但作为 IT业的百年老店，IBM在专业领域始终都是一位英气勃发的领导者：世界 500强使用率 85%以上的数据库系统，最受信赖的大型机，坚如磐石的 AIX Unix系统，庞大复杂的 WebSphere应用服务器软件……一个个沉甸甸的名字背后，是 IBM作为“蓝色巨人”的风采。老店在普通计算机用户面前是矜持的：产品一般价格不菲、安装往往冗长复杂，硬件动辄要求服务器级别，所以传统上讲普通开发者用户与 IBM往往是无缘的。不过矜持的个性没有抹煞老店的活力，在开源社区日趋活跃的时候，IBM向开源社区奉献上了一道开发工具的大餐— Eclipse，它凭着优雅的设计和精巧的实现在集成开发环境的领域里大放异彩。全世界千万级的下载次数和活跃着数百万用户的开源社区见证了这个产品为软件开发行业带来的一场免费的饕餮盛宴。

IT业百年老店在全世界也是凤毛麟角，在我们将尊敬的目光投向 IBM的蓝色徽标并祝愿她一路走好一路走远的时候，也期待国人能打造出我们自己的百年品牌。

结语

成功的软件公司还有很多很多，笔者选的这几家各有其鲜明的特色。微软的精明，谷歌的聪颖，Adobe的精益求精，Apple的浪漫，IBM的稳重强健，以您的性格，喜欢哪家多些？再换个角度来说，软件公司的成功之道可谓是不拘一格，我们自己的软件公司，找到自己的道路了么？

“SBC架构”解答了软件规格的迷思

■ 文 / 赵善中



赵善中，结构行为合一(简称SBC)架构方法的创始人，人称“SBC架构”之父，世界级的“架构教主”是也。美国电脑与资讯科学博士。经历中华电信公司，美国G.E.实验室，中山大学资管系，企业架构师协会理事长。

SBC架构的应用很广，包括：SBC硬件架构、SBC软件架构、SBC企业架构、SBC知识架构、SBC思考架构等等。

依照“SBC软件架构”的说法，软件规格应该描述出软件规格的架构，软件架构包括：(A)软件结构面和 (B)软件行为面两者，如图1所示。

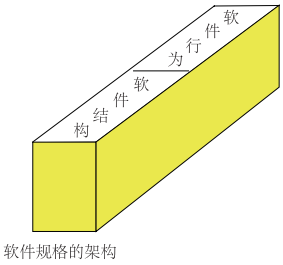


图1 SBC软件架构

过去50年来，在全世界的软件工程领域里，都错误地诠释了软件规格的方法。大家都误判为了不干涉日后软件设计与软件实现，为了讲究Information Hiding的精神，所以软件规格应该只描述软件行为面的部分。最著名的错误例子就是结构化系统分析与设计。在结构化系统分析与设计里，软件规格采用数据流程图来表达软件行为面。

这种大错特错的方法，不但在以前需求工程的英文书籍里处处可见，甚至连2009年9月出版的IEEE Computer杂志里，Daniel E. Cooke等人写了一篇文章，还在强调使用Sequence Functional Programming Language来当作软件规格语言。

我在这里必须要告知大家的是，Functional Programming Languages只注重软件行为面的部分。若是拿Functional Programming Languages来当作软件规格描述语言，是不适合的。好的软件规格描述语言是要软件结构面和软件行为面两者都兼顾到的。

在SBC架构框架里，如图2所示，软件规格的架构包括：软件规格的结构面和软件规格的行为面两者；软件设计（与软件实现）的架构包括：软件设计（与软件实现）的结构面和软件设计（与软件实现）的行为面两者。换句话说，软件规格需要描述出软件规格的架构；软件设计也需要描述出软件设计的架构。

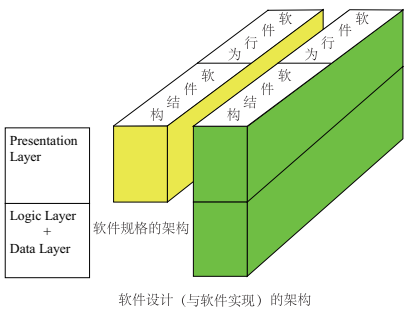


图2 SBC架构框架

“SBC软件架构”方法要求软件规格必须描述软件结构面和软件行为面。这样的要求是没有违背 Information Hiding精神的。为何如此说呢？大家可以再回去看图2所示的SBC架构框架，软件规格的架构和软件设计（与软件实现）的架构是2个架构，这 2个架构有些相关，但不违背Information Hiding的精神。换句话说，软件规格的架构对软件设计（与软件实现）的架构是充分提供 Information Hiding的精神，因为软件设计（与软件实现）的架构乃是软件规格的架构之可验证的延伸。也就是说，软件设计（与软件实现）的架构可以充分让软件设计师（与软件实现者）充分发挥Information Hiding的精神，因为软件设计（与软件实现）的架构乃是软件规格的架构之可验证的延伸。

总而论之，过去软件规格的迷思，就是没有依据“SBC软件架构”的说法，以为软件规格可以只描述软件行为面的部分，这是大错特错的方法。为了破除过去软件规格的迷思，就应该遵循“SBC软件架构”的指示，软件规格要有软件规格的架构，软件设计（与软件实现）要有软件设计（与软件实现）的架构。是故，软件规格一定要同时描述出：软件规格的结构面和软件规格的行为面两者。切记！切记！软件规格千万不能只描述软件规格的行为面，必须同时描述软件规格的结构面以及软件规格的行为面。